「情報」　試験対策プリント

編：2017 年度文１・2―14 組　情報シケタイ

このシケプリは、教養学部前期課程１年Ｓセメスターで開講される必修科目「情報」のためのものである。使用される教科書「情報（第２版）」の内容から、**必須学習項目（赤文字）**及び**要望学習項目Ａ（緑文字、文科生向け）**の内容をまとめたものである。このシケプリに掲載されていない内容は、各自教科書を読み理解することを強く勧める。

注意：現在使用している教科書「情報」は第２版であり、2016 年度以前の教科書「情報」とは内容の改変が少々行われている。

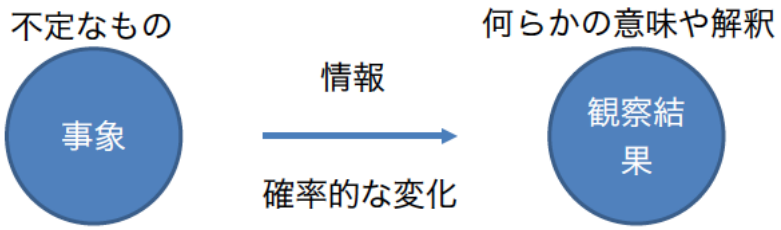
第１章　「情報の学び方」

（第１章は導入なので読み飛ばし可）

§１　情報の性質ととらえ方

○　**情報**は「形」のないもの＝無形物

「人の心（情）にはたらきかけるもの」（国語の辞書的意味）



・ある（不定な）事象を観察した時、観察した結果により、観察した事象についての「何か」が、ある確率で「明らか」になる時、そのような確率的な変化をもたらすものを「情報」と呼ぶ。

・メディア：無体物である情報の入れ物（medium:媒介物）

§２　情報の多面性

・情報には、視点の取り方、あるいは情報を取り巻く関係性の点から

○　人間に関わる側面：ヒトが情報、情報機器を取り扱う側面

○　問題解決に関わる側面

：情報機器をいかに用いるか、情報機器によりどのように問題を解決するのかの側面

○　社会に関わる側面

：ヒトの集団＝社会と、情報、情報機器、情報システムとの関係性の側面

§３　情報活動の諸要素

＜1－３－１　表現と伝達＞

○　他者に対して「何か」を伝える際

・何を表現するのか（**表現の対象**）

・なぜ表現するのか（**表現の目的**）

・どのように表現するのか（**表現の方法**）

の３つについて考慮する必要がある。

○　情報の伝達

・情報の伝達は往々にして失敗する

　→情報が正しく伝わる・間違えるってどういうこと？

・情報を正しく伝えるためにはルールが必要

・そもそも伝達が可能である条件、すなわち送り手と受け手の間に成立している共通の理解も問題

　＝その共通理解を一般に伝達のための**プロトコル**（protocol）という

＜1－３－２　モデル化＞

・情報を機械システム系で処理する場合には、計算機を用いて、数理的処理を行う

・計算機が取り扱えるような「形」にすること＝**モデル化**、数理モデル化

モデルには２種類ある

・**データ**（情報処理の対象となるもの）**のモデル**

・**計算**（計算に基づいた情報処理の仕方）**のモデル**

・計算機は、人工的な言語である**プログラム**（**プログラミング**）**言語**で記述された情報処理のやり方に従って、データを数理的に処理する。

（計算機はプログラム次第の機械）

＜1－3－3　問題解決＞

・何らかの目的（例えば、実験データの統計解析）で情報を処理する際、計算機はモデル化されたデータに対して、プログラム言語で記述された数理的な情報処理のやり方に従って計算を行うことで、目的とする**問題解決**を行う。

・この問題解決のための、計算モデル上で構築される「やり方」のことを、**アルゴリズム**と呼ぶ。
すなわち、計算機はアルゴリズムが記述されたプログラムに従って計算を行う機械である。
・プログラム言語とは人工的な言語である、つまり、アルゴリズムを人工的な言語（プログラム言語）で記述することが「**プログラミング**」である。

§４　計算の機構

＜1－4－1　コンピュータ（計算機）＞

・元々は、数学的な計算を行う（弾道ミサイルの軌道計算など）を行うために発明されたが、世の中の種々の事象や問題を、前項で述べた数理的なモデル化を行うことで、計算をベースとして問題解決の手段として用いることができる（できることがわかる）ようになったために、「万能」な情報に関する「問題解決」の「**自動機械**」となった。

・現在では、通信（ネットワーク）技術と結合されることで、極めて知的な人工的な環境（ヒトとヒトの集団を取り巻くバーチャルな環境、空間）を構成するようになってきた。

→コンピュータはどうやって情報を表現するのか？
→コンピュータはどうやって計算するのか？

＜1－4－2　2進数モデル（計算機内部での数の表現）＞

〇計算機内部では、数の表現に2進数を用いている。2進数を採用したことは、コンピュータのハードウェア（論理回路）を実現する上での重要なアイディアであった。2進数を用いることで、ハードウェアにおいて効率的な計算回路）が実現（実装）できた。

〇**Bit(0,1)**とも言う。2進法に基づく論理回路における計算の基本原理を**ブール代数**と呼ぶ。

§5　情報システムと社会

＜1－5－1　情報システム＞

〇**情報システム**：ソフトウェアを含むコンピュータなどの情報処理機器と、情報伝達のためのネットワークとを組み合わせることで、様々なサービスや機能を提供することを可能にする複合的システム

〇現在、知的な問題解決を可能とする計算機と通信（ネットワーク）技術が結合されたことで、極めて知的な人工的な環境（ヒトとヒトの集団を取り巻くバーチャルな環境、空間）が実現され、種々のサービスや機能を提供することが可能となった。このような、情報に基づくサービス、機能を提供する「**情報システム**」が、ヒトの生活、社会の活動に必要不可欠なものとなっている。

→このような状態が、いわゆる「情報化社会」と呼ばれている所以

＜1－5－2　ユーザインタフェース＞

〇　ユーザインタフェース：利用者（ユーザ）が情報システムと関わる部分

〇　ユニバーサルデザイン：広範囲の知識水準と能力レベルをもつユーザを対象としたシステムデザインのこと

→現代の情報システム開発にとっても重要な項目とみなされる、

＜1－5－3　社会＞

○　情報技術は、ヒトの生活、社会に与える影響が極めて大きいので、正しい**リテラシー**を持って、技術及び技術がもたらすものに、対応し、関係性を持ち、使いこなすことでより良い生活や社会を実現することが重要。

○　情報リテラシーにおいて注意すべきこと
・情報の真偽、確度を見抜くこと、あるいは見抜く方法論を知っていること。
・情報利用におけるリスクとベネフィットを正しく認識すること
・根拠の薄い情緒的なものに流されないこと、冷徹に論理的に考えること
・情報がもたらす利便性や利益と、情報に関わるリスクとを冷静に判断すること

○　情報と社会の関わり
・大量の情報・情報システムがもたらす社会的影響は多大
　例１）嘘の情報を発信するとどうなるか？
　例２）他人の情報を傍受・改ざんすると？
・情報に関する地検の集積・制度の整備はまだ十分でない

→各々が逐一判断し、対処する必要がある＝情報に関する「**教養**」が必要

第２章　「情報システム」

§１　情報システムとは？

情報を処理、蓄積する機器（例：コンピュータ）＋情報を伝達するネットワーク
　→これが様々なサービスや機能を提供している。
例）オンラインチケット予約、電子掲示板、POS(Point of Sales)システム
　　金融取引システム（ex.オンライン株取引）

・ICT : Information and Communication Technology = 情報通信技術
　→現在では ICT も社会インフラの一端。
以前（のイメージ）：大型計算機とその周辺機器という大規模ソフトウェア
　　関係できる人が限定されている、「聖域」的。
現在：身近にあふれる（コンピュータからスマホ、ゲーム機まで）、日常と密接。
　　→情報システムと認識されにくい。
例）スマホアプリ、情報システムが組み込まれた製品（組込システム）
　　→デジタルカメラ、エアコン、カーナビなど

§２　情報システムとしてのスマホアプリ

＜身近な情報システム＞

・情報システムを使うには？
「**導入**」**からの**「**運用**」というプロセスを踏む。
導入：ビジネス用大規模システムの場合、一般には大きな費用と手間がかかる。
　　しかしスマホアプリでは「ストア」からのアプリのダウンロード及びインストールは導入に当たるが、コストは低い。
運用：自分の使える環境下で利用すること。

・現代の情報システム
導入（ex.ダウンロード）も運用も**インターネットとの接続**が不可欠。
ローカルな処理（スマホ上など）とインターネット上のサーバ処理の共同で仕事を進める。**AI** や **AR** といった先端的な研究成果の利用も見られる。

・スマホ等モバイル端末＋ネット上サーバの組み合わせ
　→**ユビキタスコンピューティング**という形態（ユビキタス＝どこにでもある）

＜集中と分散＞　・・・情報システムの基本的構成は集中型と分散型を往来

集中型：中央集権的に全体の制御処理を司るホストコンピュータがある
　　利用者は端末からホストコンピュータに接続して情報システム利用
　　・全体の整合性を取りやすいが、柔軟性や拡張性に欠ける。
　　→銀行システムなどの伝統的システム

分散型：ネットワーク上に分散した多くのプロセッサがそれぞれ処理を行う。
　　プロセッサ同士が互いに必要な情報を交換して全体の整合性をとる。
　　・全体の整合性が取りづらいが、柔軟性や拡張性が高い。
　　→インターネット、Google のデータシステムなど

＜クライアントサーバモデル＞　・・・分散型の１つの典型例

・**サーバ**：ネットワーク上でサービスを提供するプログラム
　　（そのプログラムを搭載しているコンピュータ自体を呼ぶことも）
・**クライアント**：サーバに対してサービスを要求するプログラム

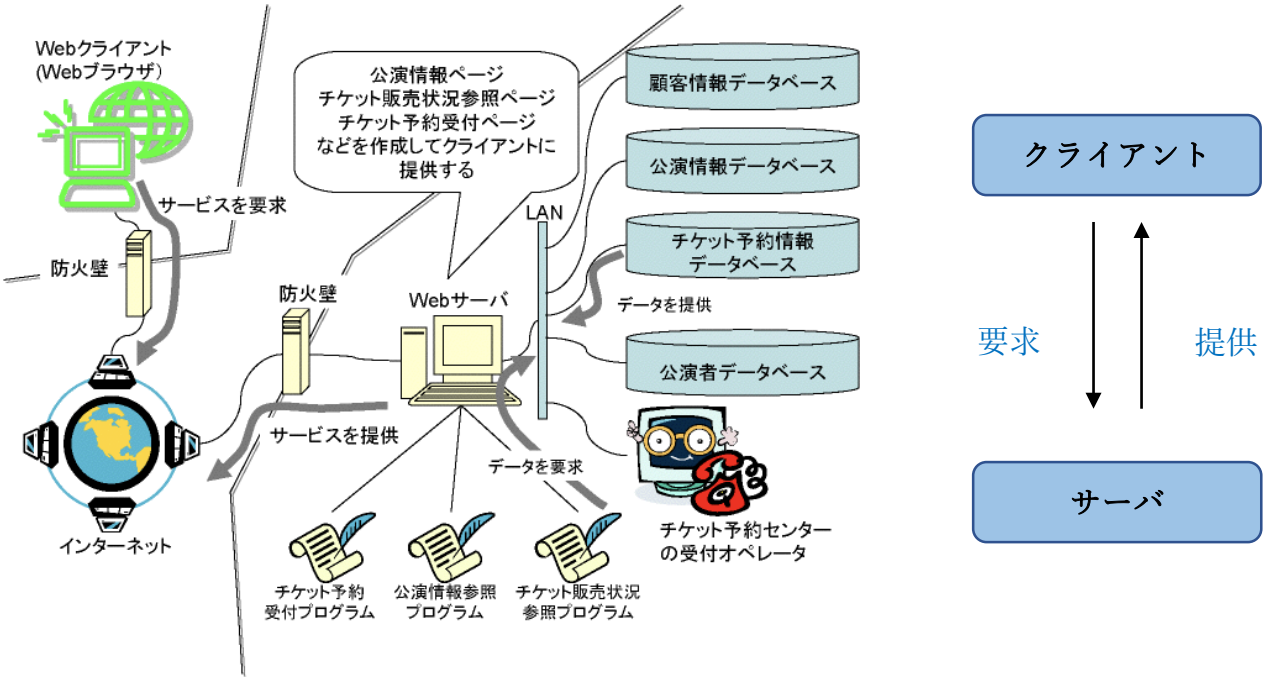


図２－１

・チケット予約システムの例　　・・・図２－１を参考のこと

① 情報照会　：ジャンル・地域・販売スケジュール

② 講演詳細情報提供

：個々の公演の情報（内容・出演者・会場・チケット情報（値段・座席）・手続き情報・主催協賛）

③ 予約機能：公演指定・チケット受け渡し方法・予約確定・決済

④ 付随機能：誤入力処理・キャンセル処理（公演キャンセル・予約キャンセル）

＜クライアントとサーバ間のやりとりと通信規約＞

・利用者が該当ページにアクセスする場合（ページの要求）

① URL で直接指定する　②見たいページを指すリンクをポインタでクリック

→このページの要求は **HTTP** と呼ばれる通信規約に従い実行される。

・用語解説

HyperText　：参照されているテキストを自動的に参照する仕組みを備えたテキスト
Web：ハイパーリンク（HyperText から HyperText へのリンク）で相互結合された文書群
WWW（World Wide Web）：インターネット上に構成された Web
URL (Universal Resource Locator) :インターネット上の資源特定のための形式的記号表記の仕方

● HTTP(Hyper Text Transfer Protocol)

→クライアントとサーバ間のやりとりに関する通信規約（プロトコル）の一種

(端的にいうと HTML ファイルなどを渡すための決まりごと　ということ)

通信要求のコマンド例

・**GET**　：クライアントがサーバから情報の資源を取得するために出す要求

（GET により、web ページなどを取り出せる）

・**POST**：クライアントがサーバに情報を与えるために出す要求

（web フォームでの情報入力や掲示板への投稿などに使用）

＜クラウドコンピューティング＞

クラウド：インターネットの向こうにあり、直接意識しない
　　ような見えない情報
　＝「雲」のかなたにあるものとして考えられる

・利用者は自分の環境に導入する必要なし。
　インターネットを介してサービスを利用できる。
　近年クラウドの利用が増えている。

（参考）クラウドコンピューティングの形態

・SaaS：利用者向け。ソフトウェアを購入せず、ウェブを介してサービスとして利用。
・DaaS：利用者向け。データをクラウドで利用。特に利用量が多い。
・PaaS：開発者向け。プラットフォームのサービス利用。

§３　ビッグデータ

・ICT で扱うデータ量の巨大化、データを処理する処理システム及び通信システムの能力と容量の巨大化→データの巨大化＝ビッグデータ（定義が少し難しい）
・ビッグデータの技術により以前では捨てられていたデータが使われたり、データをより生のまま使う傾向が現れている。

ビッグデータの発生源

・ウェブページ　（SNS、動画サイト、写真サイトなど）
・通信　（メール、電話、チャット、SNS のコメント）
・社会行動　（IC カード、POS 情報、位置情報、決済情報など）
・センサー　（検出データ、※IoT）
・科学データ　（観測・実験データ）

※ IoT（Internet of Things）・・・モノのインターネット

第３章　情報の表現　　―記号・符号化―

§１　情報の表現

＜“表現”のさまざまな側面＞

● 情報を表現する言語には違いが存在する。

・**自然言語**：我々が特別な訓練なしに日常的に使用している言語　ex）日本語
・**人工言語**：人工的に作られた言語　　ex）プログラミング言語

● 情報の説明の仕方の違い

・**手続き的表現**：時間を追った手順を説明
・**宣言的表現**　：対象間の関係や対象の属性を説明

例）ある場所を人に教えるとき

「二つ目の信号を右に曲がって三軒目」・・・手続き的表現

「三越デパートの隣」・・・宣言的表現

● 情報の表現のされ方の違い

・**記号表現**　　：与えられた記号の集合と解釈するための規則体系 （例：数学の記号、数式、方程式、論理式）
・**パターン表現**：構成要素間の時空間的パターンによる表現（例：地図）

その他にも　・ディジタル/アナログによる表現の違い（→§３）

・情報量から見る側面（→§５）　　など情報の表現は複数存在

＜情報の表現とモデル＞

・**モデル**　：単純化、抽象化された事物/事象/概念
・**モデル化**：実際の事物、事象に対応したモデルを構築する過程のこと

モデルの例：（ジェット旅客機の設計）

→実際の旅客機をテストする前に小型模型（モデル）を用いた風洞実験を行う

モデルの表現形式・・目的によって構築するモデルの表現形式が異なる

風洞実験→実機と同様の材料でモデル化

デザイン目的→加工、修正しやすい材料でモデル化

＜モデル化の表現形式の例＞

(a)　表（table）

最も身近なモデル化の表現形式の１つ

・こみいった事柄を整理できる

・歴史年表/貸借対照表/成績表など

・計算機上の表計算ソフトが一般的に利用されている

(b)　図

表と並んで最も日常的に用いられる表現形式の１つ

・何らかの目的で描いたすべての２次元図形を指す

・こみいった事柄を整理し、人間の思考・推論を支援/拡張する

・設計図や地図など。広義には絵画やスケッチなども含める

(c)　グラフ

グラフはモデル化において非常によく用いられる表現形式である。

・**ノード(node)**と**エッジ(edge)**から構成される(ノードは**頂点**とも呼ばれる)

・**ラベル付きグラフ**・・・ラベル付きのエッジで構成されるグラフ(→図３－１)

・**有向エッジ、弧**・・・方向を持つエッジのこと

図 3―1 のような道路ネットワーク以外に、組織内の部署の関係（**組織図**）や工場における作業工程（**PERT 図**）、人言の記憶構造の表現（**意味ネットワーク**）など幅広く用いられている。

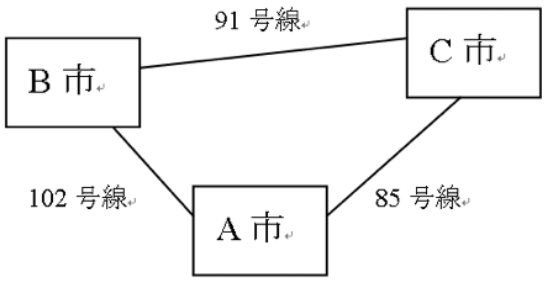
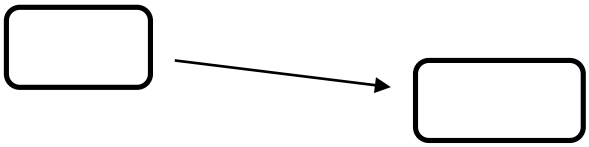


図 3－1

図 3－1 においては

・四角で囲まれた部分＝ノード

・直線＝エッジ（ラベル付き）



↑有向エッジ

＜情報の表現とは＞

● 情報表現で考慮すべきこと

・情報表現の受け手側　：情報表現を適切に理解・解釈しなければならない

・情報表現のデザイン側：目的に応じて適切な表現手段を選択する必要性

・情報は解釈・処理されるものである

→　特に人間が扱う情報：解釈される　/ 特に機械が扱う情報：処理される

● 情報表現を解釈・デザインする際に考慮すべきこと

（a）　表現の対象

・表現の対象となる事物/事象を明確にする必要がある

→表現の対象：物理的実体、抽象的アイディア、記憶的事象、思考の方法・結果

（b）　表現の目的

・表現されている/する目的を理解する

→表現の目的：他者への伝達や依頼、自身のアイディアの整理、効率的問題解決

（c）　表現の方法

・表現に関わるコストや目的に照らし、より良い方法を選択する必要がある

§２　記号と表現

＜記号表現＞

・記号表現：事物・事象、心的概念を抽象化したもの

・記号表現の実際の形式：図記号（ピクトグラム）、数の表現など

・記号が表す２側面　→サービスエリアを例にとる。

：意味されるもの・表されているもの（シニフィエ）　（例）サービスエリア

：意味するもの・表しているもの（シニフィアン）　（例）図記号

図３－２

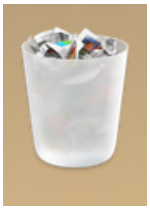


図３－３

＜図記号（ピクトグラム）　―記号と意味＞

・図記号の修辞法

● 提喩

・全体と部分の関係で構成された比喩。全体の呼称を提示して一つの名称に変える。
例）「花」で「さくら」を表す類、「パン」で「食物全体」を表現する類

・提喩に相当する表現方法

ある事物を表現するのに、それと意味的包含関係にある事物を代わりに用いる比喩。図 3－2 では「ナイフ、フォーク」の図で「サービスエリア」を表現

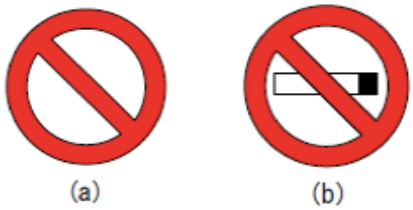
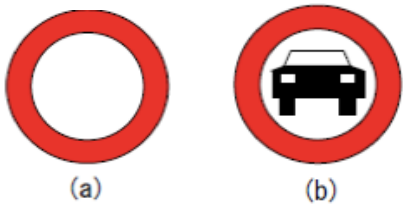
● 隠喩（＝暗喩）

・例えを用いながら表現面にその形式を表さない（「ごとし」「ようだ」等を用いない）比喩。

・隠喩に相当する表現方法

GUI におけるゴミ箱アイコンは「ゴミを捨てる」という行為の隠喩　→図 3－3

・交通標識の図表現

	
＜日本＞ （a）　車両通行禁止の標識 （b）　禁煙の標識	＜欧州＞ （a）全ての車両通行禁止（欧州） (b)二輪車以外の車両通行禁止（欧州）

●記号の恣意性

・記号表現と命題の対応付けは恣意的である
→日本では当然の〇、×による表現が常に肯定、否定（禁止）に対応づけられるわけではない
・情報表現のデザイナーは受け手側の解釈の枠組みに注意を払う必要がある

● 文字の符号化

・例えばアルファベット（ラテン文字）を含む文字集合をビット（７bit）の数値は **ASCII**(American Standard Code for Information Interchange) で対応づけている。

〔日本語文字コード〕

・文字と計算機上の符号（数値）を対応づけるための枠組み

・現在 **UTF-8、JIS、シフト JIS、EUC-JP** などの異なった日本語文字コードが混在して使用されている。

・解釈の枠組みが異なれば記号の意味が異なってしまう例

→プログラムが想定するコード体系と異なると「**文字化け**」が起こる。

・コード体系の標準化・統一化には困難も多い

→歴史的経緯、利害関係、処理の都合、拡張性、文字セットの違いなどが要因

〔文字の符号化の問題〕

・歴史的性質：過去に符号化された文字は読めるべき

・転送・記録の効率：短い符号化→早く転送・たくさん記録

・文字の量：ヨーロッパ語は数十文字・漢字は数千文字以上

・複数の標準：メーカーごと、地域ごとに決定

・細かな、しかし文化的に無視できない違い

：見た目の類似性、異体字、方言ごとに異なる文字

・国際化：狭いコミュニティだけの使用→世界中のコンピュータが通信をする時代、多言語の同時使用

＜数と表現　―記号と解釈の規則体系＞

● 数の表現の歴史

・インドでの発見

→数字の“ゼロ（０）” 、位取り表記法による算術演算

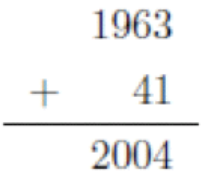
・アラビア数字表記法

→０～９の１０種類の記号の使用

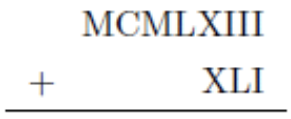
各記号が各桁に対応している

・ローマ数字表記法
→Ⅰ、Ⅱ、Ⅲ、Ⅹ、Ｃなどの記号を用いる

● 位取りに基づいた計算
・アラビア数字における **1963＋41** の計算
→表記上の各桁を計算していけばいい



・ローマ数字における **MCMLXⅢ＋XLⅠ** の計算
→位取りに基づく計算をすることができない



● **情報表現間のトレードオフ**
（→ある側面を優先させると別の側面に問題が生じてしまう関係）
・アラビア数字表記
→筆算や位取りの観点からは優れている
・ローマ数字表記、漢数字表記
→表記された数字の改ざん防止に優れている
・アラビア数字表記「2030」は「1203000」のように改ざんされやすい
・漢数字表記「弐千参拾」は改ざんされにくい

→情報表現のデザイナーは情報表現間のトレードオフを考慮する必要がある

● **コンピュータでの数の表現**
→「０」と「１」の２種類の記号を用いた**ビット列**で表現される
例）　１０進数の５は"１０１"、１０進数の７は"111"　と表現される
・表現できる数値はコンピュータによる
→１６ビットのシステム：０～65535（＝$2^{16}-1$）までを表現できる
→３２ビットのシステム：０～4294967295（＝$2^{32}-1$）までを表現できる

・数値といっても色々ある（整数・有理数・少数点数・複素数など）
・「表現」する際にはその精度が問題となる
→無限に大きな数、無限に細かい数を「書き表す」には無限の文字が必要
ここで、まずは有限の正の整数を考える

● 正整数の表現：２進数
・コンピュータで最もよく使われている表現
・基数を２とする（単純で信頼できる）→各桁は０と１だけになる
・位取り記法を使う（計算に便利）→n 桁目は 2^n 倍された数を表す

［２進数の性質］
・n 桁の２進数で表せる範囲は０～2^n－１まで
例）８桁の２進数→０から２５５まで

・１０進数と同じ方法で四則演算ができる（位取り法を使っているため）
（例題）２進数の計算問題

［数値以外の情報の表現］

［情報量の単位］
・最も基本となる（最小の）単位は**ビット　→２進数の１桁**に相当する
・１バイトは**８ビットに相当する**
・（問題）正しい選択肢を選べ
　→日本語の漢字１文字は（１）１バイトで表現される
　　（２）２バイトないし３バイトで表現される
　　（３）５バイトで表現される
　　（４）10 バイトで表現される
　　（５）適切な選択肢はない

→正解は（５）

§３　アナログとディジタル

＜アナログ表現とディジタル表現＞

● アナログ表現

・ある情報を連続量（**アナログ量**）として表すこと

・無限の精度を必要とするため、データの複製は元のデータの近似にしかならない→データの劣化が避けられない

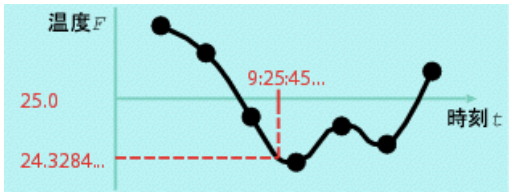
● ディジタル表現

・ある情報を離散的に表すこと（ディジタル量）

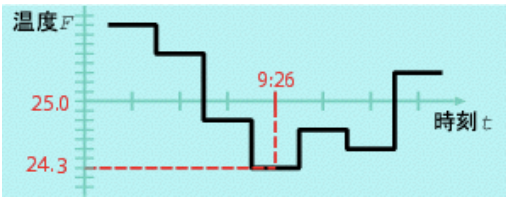
→ある情報に対して一定の間隔の尺度を導入し、その尺度の値に近似して表現する

・複製時にデータが劣化しにくい

・情報コンテンツの著作権保護への問題をもたらす（→第 10 章の§３）



（左）気温のアナログ表現



（右）気温のディジタル表現

・アナログ量をディジタル量に変換する際には、情報を離散化する間隔を選択し、表現する必要がある（もともとの連続的な情報が切り捨てられる）

・アナログ量→ディジタル量変換では**標本化**と**量子化**の２つの作業が必要

　・**標本化**　：一定時間間隔ごとの計測（抽出）すること

　・**量子化**　：測定値をある間隔ごとに表現すること

＜量子化＞

量子化：連続量の情報を有限個の段階の離散量として表現すること

→段階を多くすれば、より詳細な情報となる

・情報の用途によって間隔の詳細度を決める

例）コンピュータディスプレイ装置のカラー表示

・赤(R) 青(B) 緑(G)を混色した RGB 形式を用いている

・各々２５６種類（＝8 ビット）の異なる色で表現

・256×256×256＝16777216 色（＝24 ビット）を表示できる

音楽 CD

・音の振幅を 65536（＝2^{16}）個の段階に分割している

・65536 段階は 16 ビットで符号化される

＜標本化定理＞

標本化：情報をある間隔（頻度）ごとに抽出すること

例１：温度を「１時間ごと」に測る

例２：部屋の温度を「１０cm」に測る

例３：音圧を「0.0000227 秒ごと」に測る

標本空間：標本化の対象の情報が定義されている時間や領域

例）　・ある音楽が鳴っている時間

・ある絵画全体の領域

・標本化の間隔はデータの利用目的により変わる

→・どこまで細かく見たいか

・間隔を広げてゆけばデータの量が減る→同時にデータは「荒く」なる

→では何を失うのか？

・高い精度が求められるのか、荒くでもサイズが小さい方が良いのか？

§４　ディジタル符号化

・符号化：ある情報を他の記号集合で表すこと

＜ディジタル符号化の例＞

・**2 進符号**：１０進数を２進数に変換したもの（０と１の記号による表現）

→１０進数なら$(1)_{10}$ 、２進数なら$(1)_2$　と下付き添え字を用いて区別する

・**ハミング距離**：２つの符号間で対応する桁の記号が異なる個数

すなわち、ある文字列を別の文字列に変形する際の置換（記号置き換え）回数

→・（0000）と（0001）ではハミング距離は **1**

・（0011）と（0100）ではハミング距離は **3**

・2 進符号では数値の差とハミング距離が一致しない

例：（0001）と（0101）では数値の差は５だが、ハミング距離は１

・**グレイ符号**：値が隣接する符号間のハミング距離を常に１とした符号

１０進数	2進符号	グレイ符号	１０進数	2進符号	グレイ符号
0	００００	００００	8	1000	1100
1	０００１	０００１	9	1001	1101
2	００１０	００１１	10	1010	1111
3	００１１	００１０	11	1011	1110
4	０１００	０１１０	12	1100	1010
5	０１０１	０１１１	13	1101	1011
6	０１１０	０１０１	14	1110	1001
7	０１１１	０１００	15	1111	1000

・グレイ符号は２進符号から作ることができる

→・最上位桁は２進符号と一致

・最上位桁以外では、対応する２進符号の桁とその左の桁が一致すれば０

異なっていれば１となる。(←実際にやってみたがよくわからない・・・)

・グレイ符号はパターン生成や機械の制御コード、遺伝子の変異を模した計算などにも用いられる

（偽のデータが作られにくいという利点もある）

＜ディジタル符号化の特徴＞

● ディジタル符号化の応用

・圧縮

→・情報を失わずにデータの量を減らす

・表現される情報の特性を利用する

・誤りの検出と訂正

→・検出：雑音などで誤ってしまったことを発見する

・訂正：雑音などで誤ってしまった情報を元に戻す

● ディジタル符号の圧縮

→ディジタル符号化された情報は圧縮できる利点を持つ

・**可逆圧縮**：圧縮したものから元の情報を完全に復元できる方法

→PNG や GIF などの画像データフォーマットは可逆圧縮形式

・**非可逆圧縮**：元の情報には復元できない圧縮方法

→画像データフォーマットの１つである **JPEG 圧縮**

JPEG 圧縮・・・要求される精度の周波数成分までを符号化する

→人間が必要としない高周波成分に対する情報を切り落とすことでデータの高い圧縮率を実現

・非可逆圧縮は人間の近くでは差異がわからない程度の復元が可能ならば様々に応用が可能

● 符号の誤り検出・訂正

・ディジタル表現はアナログ表現に比べて情報伝達時の誤り検出や訂正が容易

・ディジタル情報は通信時にノイズが混入しても情報を正しく復元することができる

→符号の誤りの検出・訂正について実際に例を出して考えてみる

〔A、B という情報を相手に伝えたい場合にノイズによりビットが１つ反転しうると仮定すると〕

①　A：0、B：1として符号化した情報を伝送

→「0」という情報はAから来たものか、Bの情報が反転したもの？？

→受け手側では誤りを検出できず、訂正すらできない

② A：００、B：１１　と同じ符号を重ねて符号化した2ビットの情報

→受信する可能性のある符号は4通り（００、０１、１０、１１）

→誤りを検出できる・・01、10は誤りが生じたものだとわかる（同じ符号が重ねられてはいないため）

③ A：０００、B：１１１として3ビットの情報として符号化

→受信する可能性のある符号は8通り（→000,001,010,100,011,110,101,111）

→この場合も誤り検出と訂正が可能である

・001、010、100はAを伝送しようとして誤ったもの

・110、101、011はBを伝送しようとして誤ったもの

§５　情報の伝達と情報量

＜情報の伝達＞

● **情報の伝達**：受け取り側の状態変化が本質

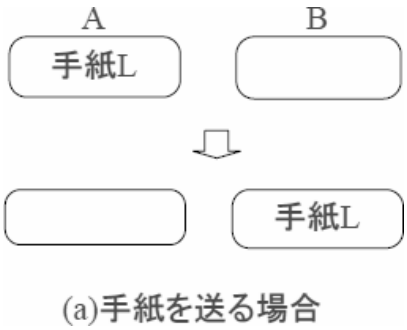
・様々な伝え方で同じ「情報」（メッセージ）が伝わる

→・「手紙」を送る / 「手紙のコピー」を送る

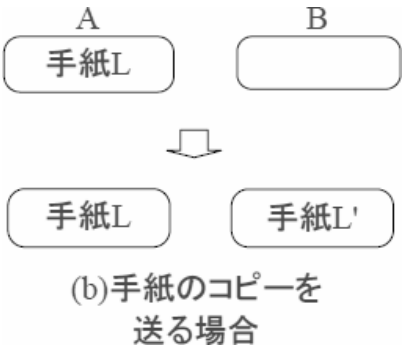
・電子メールを送る

→手紙の物理的な移動は本質ではない

同じ内容を受け取る限り、電話でも電子メールでも同じ効果がある



(a)手紙を送る場合



(b)手紙のコピーを送る場合

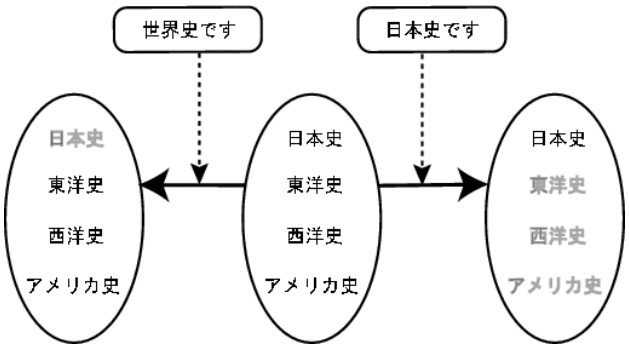
・B さんが受け取る情報はどちらの場合も同じ
・物理的な手紙の移動は無関係
　→A さんの手元から手紙が消えることは本質ではない

● **情報量**　：情報を受け取った効果を測る
・情報を受け取った効果についての直感的な説明
→情報を受け取った場合
　：自分に影響がある、これまで知らなかった事実を知った
　　何らかの判断材料にできる事実を知った
→情報を受け取ったと言い難い場合
　：関心のない手紙を受け取った（例：迷惑メール）
・情報を受け取る効果は、受け取る人の「状態」と関係がある
・メッセージの効果を「**情報量**」として表現したい

・情報の価値は受け手の選択肢をどれだけ減らすかで測る
　→以下の試験に関する例をもとに考えていこう！

・科目「歴史」の試験
　・日本史、東洋史、西洋史、アメリカ史のどれか１つが出題される
　・事前にはどれが出題されるかは分からない
・メッセージ：「今回は日本史から出題する」
・状況の変化
　事前：日本史からアメリカ史の４種類全部の試験勉強が必要である
　事後：日本史の勉強だけで済む

※ここで、仮にメッセージが「世界史から出題する」の場合、出題に関する曖昧さは４→３となるだけであり勉強がしやすくなる度合いはそれほど大きくない



＜情報の大きさ　－情報量＞

(a) 場合の数の変化

前置きとして・・・情報量の決め方について（場合の数をもとにする）

案	定義	問題点
1．差	（事前の場合の数）－（事後の場合の数）	100→97　の場合と　４→１　が同じ価値か？
2．商	（事前の場合の数）÷（事後の場合の数）	情報量の加法性（後述）を満たさない
3．商の対数	log(事前の場合の数 / 事後の場合の数)	

※情報量の加法性について

・情報を一度に受け取った場合（A）

　・メッセージＡ：「アメリカ史を出題する」

　　場合の数は　４→１

・情報を分割して受け取った場合（B＋C）

　・メッセージＢ：「世界史を出題する」

　　場合の数は　４→３

　・メッセージＣ：「東洋史と西洋史は出題しない」

　　場合の数は　３→１

・**情報量（A）＝情報量（B）＋情報量（C）**としたい　←**情報量の加法性**

［場合の数に基づく情報量の定義］

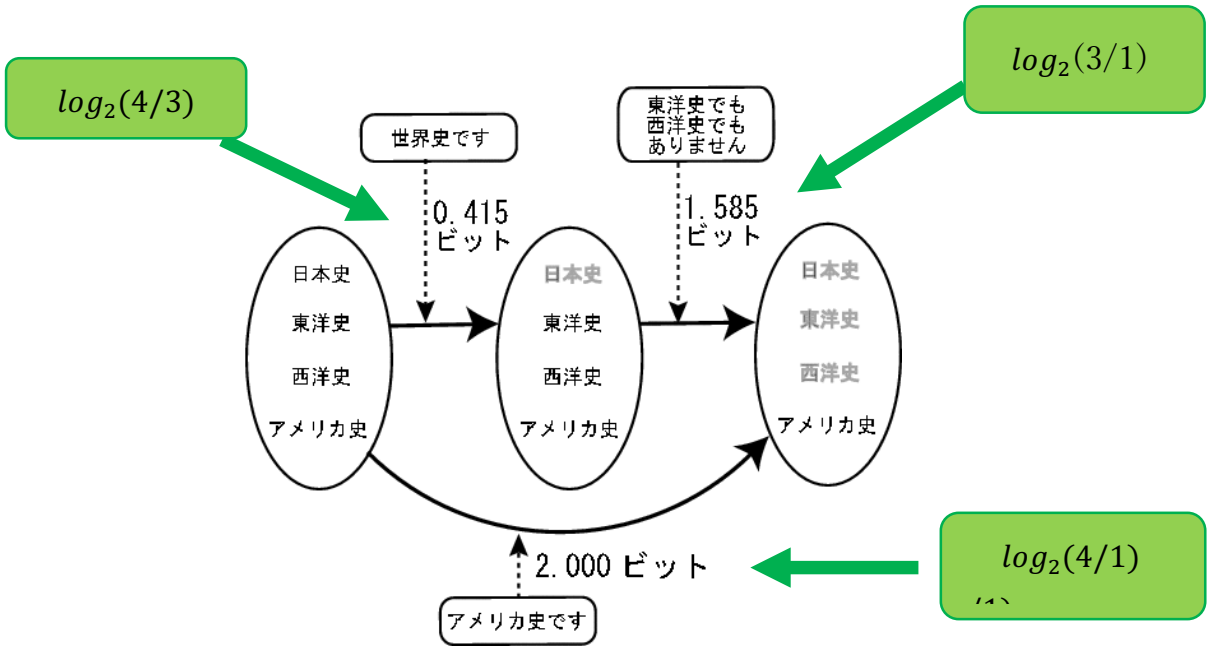
・**定義：　log_2(事前の場合の数 / 事後の場合の数)**

・**単位：ビット（bit）**

・**性質**：・場合の数が大きく減るほど数が大きくなる

・底が２であるため二者択一（場合の数が２から１になる場合）1.0

・情報量の加法性を満たす



(b)　確率による定義

以後、あるメッセージが発せられる確率をｐとする

・定義：　$log_2\left(\frac{1}{p}\right) = -log_2(p)$

・単位 ： ビット(bit)

・性質：確率が低いことを伝えるメッセージほど値が大きい

・確率 1.0→情報量 0　　・確率→情報量　1.0

・確率→情報量　2.0　　・確率→情報量　無限大

→「犬が人間を噛んだ」はよくある（確率大）ことなのでニュースにはならない

「人間が犬を噛んだ」は珍しい（確率小）のでニュースになる！（情報量大！）

・場合の数に基づく定義の一般化

：全てが等確率で起こるときは、場合の数の定義と同じ

＜平均情報量＞

個々のメッセージの情報量が定義できたところで、次は「メッセージ全体」が持つ情報量について考えてみる

→そのために個々の情報量×メッセージの起きやすさ（確率）の総和を取る

（定義）平均情報量＝ $-p_1 \times log_2(p_1) - p_2 \times log_2(p_2) - \cdots - p_n \times log_2(p_n)$

先ほどの例をもとにして考えていく

・出題範囲に関するメッセージが「日本史」「東洋史」「西洋史」「アメリカ史」のどれかである場合、これらの確率はみなすべて等しく 1/4 であるので

平均情報量 ＝ （1/4 × $\{-log_2(1/4)\}$）× 4 ＝ 2

→等確率の状況では平均情報量は個々のメッセージの情報量と同じになる

・では等確率ではない場合は・・・？

例）メッセージが「日本史」か「世界史」の２種類の場合

平均情報量 ＝ $\{1/4 \times (-log_2(1/4))\}$ ＋ $\{3/4 \times (-log_2(3/4))\}$

＝0.25×2＋0.75×0.415

＝ 0.８１１

§６　情報通信のモデル

＜符号化と平均情報量＞

・情報は０、１の符号で表され、伝達される

・伝達速度が一定なら、小さいデータほど小さいデータほど早く伝送できる

　→データは復元可能なように圧縮して伝送する

・例）2 年分の試験出題情報を符号化する場合、珍しい情報には長い符号を、珍しくない符号には短い符号を割り当てる

ここで（**定義**）　**●平均符号長＝** $\sum_i(p_i l_i)$

※ p_i：情報 i が起こる確率、　l_i：情報 i の符号長

※ 符号長・・・　符号 11 なら符号長は２、符号 010 なら符号長は３

→先ほどの例にもどって考えてみる

出題	確率	符号	符号長
日本史＋日本史	1/16	111	3
日本史＋世界史	3/16	110	3
世界史＋日本史	3/16	10	2
世界史＋世界史	9/16	0	1

ここで平均符号長を計算すると　**平均符号長＝１.6875**

→1 年分の試験情報を符号長１で表した時の、2 年分の試験情報の符号長（＝2 ビット）よりも平均符号長が短くなっている

→一つ一つ符号化するよりも、まとめて符号化した方がいい

第４章　　「情報の通信と伝達」

§１　１対１の通信とプロトコル

・プロトコル（protocol）

＝ 通信の意図を理解するための（送り手と受け手の）決め事

例）電話「もしもし」、トランシーバ「（自分の発言後に）どうぞ」と言う

＜ 4-1-1 階層化と相互運用性＞

・プロトコルの階層化

（まずは階層化について理解しましょう）

例）郵便将棋・・・離れた場所に住む相手と将棋を指す方法

→自分の指し手を一手ごとにハガキに書いて送る

・必要な約束事

① 将棋のルール・・・上位層のプロトコル

② 郵便のルール・・・下位層のプロトコル

・・・これらのルールは別々に考える（階層化）

→各階層を**独立に**考えることができる

・下位層の指し手を送るための通信手段を電話に替えても
上位層の将棋のルールは変更する必要がない

・下位層の郵便は上位層を囲碁にした場合にも使うことができる

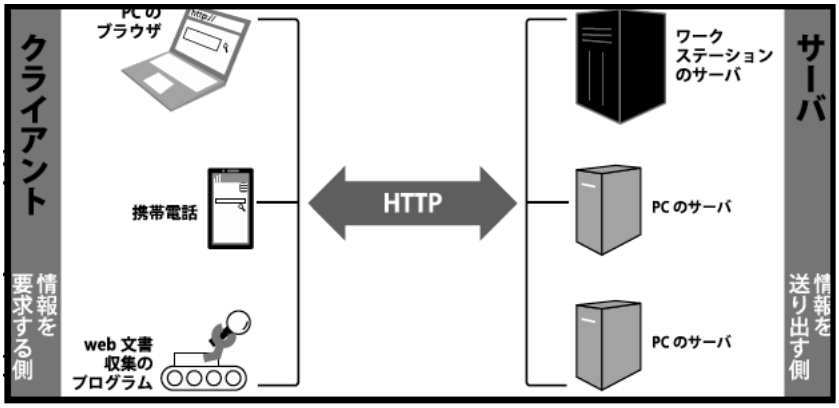
・**コンピュータの場合**（プロトコルの階層化）
コンピュータ同士の通信：人間より厳密
プロトコルを正しく使えば機器によらず通信が可能
↓

・インターネット通信における階層化されたプロトコル
上位層・・・**HTTP：ブラウザとウェブサーバの間**の通信（後述）
下位層・・・**TCP / IP** の通信　（→§２）

＜4－1－2　HTTP：ウェブのプロトコル＞

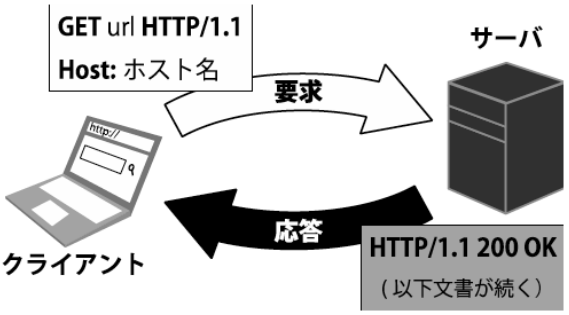
・ユーザがウェブのハイパーリンクをクリック
→　ブラウザにウェブ文書が表示される
ブラウザ・・・URL で指定されたウェブ文書をウェブサーバに要求
ウェブサーバ・・・要求に応じて適切な情報を送り出す
→これらの間のやりとりを規定するプロトコルが
HTTP (Hyper Text Transfer Protocol)　（←第２章参照）

・クライアントサーバモデル　（第２章§２も参照）
クライアント　・・・情報を要求する側
サーバ　・・・要求に応じて適切な情報を送り出す側
→先ほどの例でいうとブラウザ＝クライアント、ウェブサーバ＝サーバとなる
→　クライアントが何であっても（PC、携帯電話、検索エンジンのためにウェブの文書を集めて回っているプログラム）、HTTP で通信可能ならばサーバから文書を得られる。



HTTP：**WWW クライアント**と**サーバ**の間の通信に関する約束事

要求と応答を基本とする。



・クライアントからのリクエスト①

（１）クライアントからサーバに **GET** という文書要求を示す語句（メソッド名）と目的文書の URL を送信する

（２）サーバが応答を返す

（ページを表示するなど）

→サーバの応答の冒頭には、ヘッダと呼ばれるメッセージが付されており、応答の種類の把握を容易に。

・要求が成功した場合　→　**HTTP/1.1 200 OK** などと書かれる。

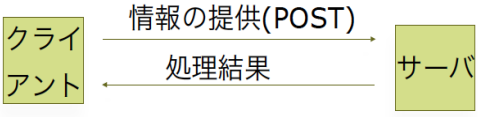
・URL に対応する文書が存在しなかった場合　→　**404 Not Found**

（他にも 400 Bad Request 「不正な要求です」　などと様々に存在する）

・クライアントからのリクエスト②

（１）GET の他に **POST** という、クライアントからサーバに処理を依頼するのに情報を送信するためのメソッドもある

（２）サーバが処理結果を返す



・**Cookie**：クライアントを識別する符号

→・クライアントから先ほどの（１）の時点でサーバに送られる

・サーバが以前の応答でブラウザに提供した符号

・Cookie の活用例

→・同じ URL であっても個人ごとに異なる内容を見せたい場合

（ログイン時など）

・閲覧履歴に合わせた広告を表示したい場合

＜4-1-3　HTTPS ：　安全な通信＞

・HTTPS　(= HTTP over TLS)
　：通信の安全に関する配慮が必要になり HTTP を拡張したプロトコル
　① やりとりされるデータが盗聴されない（データの暗号化によって）
　②　利用者や運営者の「なりすまし」の防止（**認証**を用いて）

[認証の仕組みについて]

フィッシング(phishing)：偽のサイトを立ち上げて利用者の個人情報を盗むことでは利用者をフィッシングから守るための HTTPS の仕組みは？

→**認証局**が署名したディジタル証明書でサーバの身元を確認

・・・このような信頼の仕組みを　**PKI (Public Key Infrastructure)** という

○ 安全な通信を確立するために

・運営者の立場で顧客のなりすましを防ぐために
　→　利用者コードとパスワードを正しく入力できるかを確認する

しかし、総当たり攻撃・推測・事故による漏えいなどで無力化する可能性あり

・安全性をさらに高める仕組み
　・生体認証（指紋、虹彩など）
　・２段階認証（携帯電話などを利用）
　・ワンタイムパスワード
　・ユーザ認証にもディジタル証明書を用いる（確定申告などで）

§２　インターネット

＜4－2－２　ネットワークの集合体と通信＞

・**インターネット**＝小規模な局所ネットワーク同士を繋いだネットワーク
・インターネットで通信をする場合は、情報は色々なネットワークを渡り歩いてゆく
・個々のネットワークの中では、コンピュータ同士が直接通信できる
・**ルータ**　：ネットワーク同士をつなぐ機器
　　ネットワーク間の通信を中継

● ネットワーク内外の通信

ネットワーク内通信　・・・通信媒体（無線 LAN、有線 LAN）毎で異なる
ネットワーク外通信　・・・TCP/IP という共通のプロトコル群を利用

［ウェブブラウザでウェブサイトを閲覧する例］

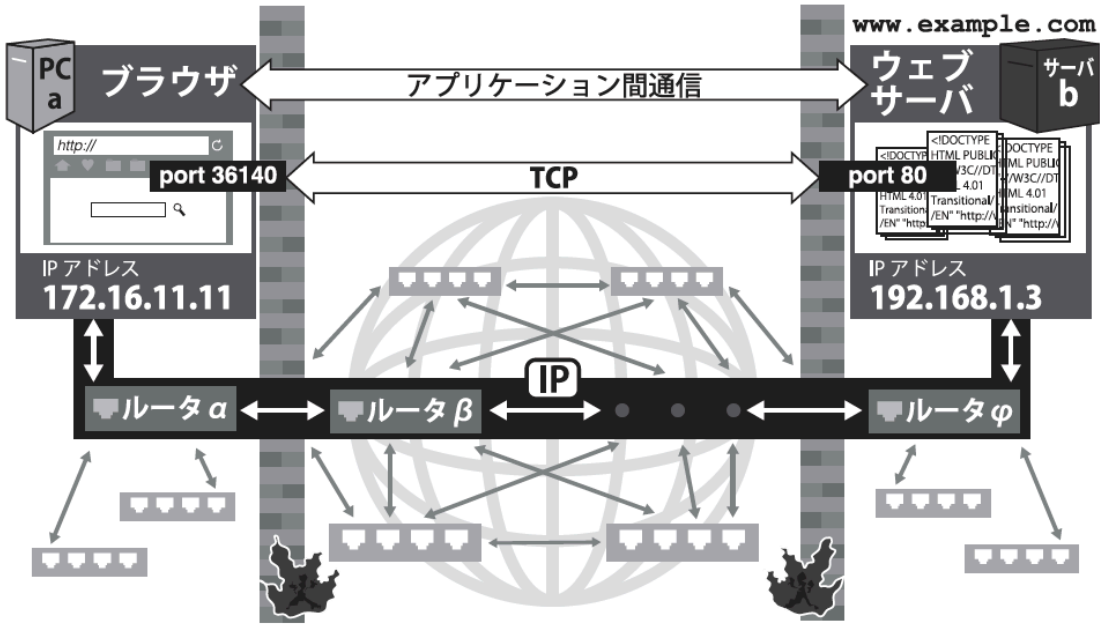


図　４－４

①　ブラウザが URL（http://www.example.com）から、（ウェブサーバの宛先である）**IP アドレス**（192.168.1.3）を **DNS**（→後出）で調べる
②　PCa から宛先の IP アドレスのウェブサーバに対して、**TCP** の（仮想的な）通信路を開く。（IP プロトコルで TCP の通信路を通じでメッセージを届ける）

（a）　PCa：メッセージをパケットに分割し、パケット毎に IP アドレスを添え送信

（b）各パケット：PC→ルータ α →ルータ β →・・→ルータ ϕ →サーバ b へ転送

（c）サーバ b：受信したパケットを並べてデータを結合、元の HTTP メッセージを取り出し、サーバで動作中のウェブサーバに渡す

※　パケット：データをを一定の大きさに小分けしたもののこと

③　ウェブサーバからブラウザへ同様に返信する

・TCP/IP は階層的なモデルに基づいたプロトコル群であり、４階層から成る。

TCP/IP モデル	主なプロトコル	宛先の主な指定方法
アプリケーション層	HTTP, SMTP, DNS, DHCP	URLなど
トランスポート層	TCP, UDP	IPアドレスとポート番号
インターネット層	IP	IPアドレス
ネットワークインタフェース層	（イーサネット）	MACアドレス

［上位のプロトコルは下位のプロトコルを通して実現される］

・**HTTP** の通信：ブラウザとウェブサーバが直接通信する

→　一つ下のトランスポート層であるプロトコル **TCP** において実現される。

・**TCP** の通信：必要に応じデータをパケット分割し、組み立て

→　インターネット層の IP プロトコルでそれぞれを配送

・**IP** プロトコル：網の目のように接続された多数の通信機器から適切な配送ルートを見つけ出す**経路制御**

→　隣接するルータ同士などで同一のネットワーク内の通信はネットワーク通信（ネットワークインターフェース）層が担当する

・ネットワークインターフェース層：優先や無線など物理媒体ごとに異なる。

＜4－2－3　ホスト名と DNS＞

DNS（Domain Name System）：インターネット上の位置の示し方

・**ホスト名**：人間に使いやすい名前（英数字など）
　→**URL**（例えば www.u-tokyo.ac.jp）など

↕ **DNS （Domain Name System）**が両者を対応づける

・**IP アドレス**：コンピュータの扱う住所（数字）　（←例えば 192.168.1.3）

・ホスト名と IP アドレスの対応は全世界共通である必要があるがデータは巨大であるし、更新の頻度も高い
　→全体を構造的に分割して、ドメインと呼ばれる一定の範囲毎に**分散管理**！

・**分散管理**　：　ホスト名は**木構造**
例）**www.u-tokyo.ac.jp**
・一番最後の jp は国名や net など決められた名前だけが入れられる。
・ac.jp　は日本の大学に当てられるドメイン

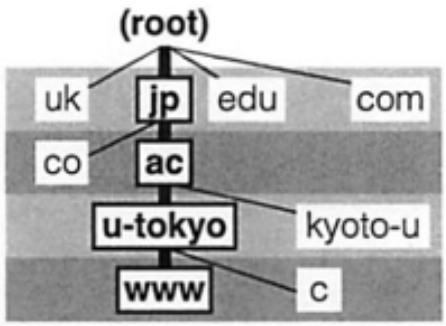


図 4.7　ドメイン名の木構造

→・階層ごとに限られた情報を管理
　・**反復問い合わせ**による解決
（参考）反復問い合わせ：木のルート（根）→葉へ
→最初はルートサーバと呼ばれる基幹サーバに jp を担当する DNS サーバを尋ねる。次に jp のサーバに ac.jp を担当する DNS サーバを尋ね・・・（以後繰り返し）・・最終的に u-tokyo.ac.jp を担当する DNS サーバから www.u-tokyo.ac.jp の IP アドレスを獲得するという仕組み
　・問い合わせの結果の再利用（**キャッシュ**）

＜4－2－4　IP アドレスとネットワークアドレス＞

・**IP アドレス**：インターネット内の住所

・32 ビットの数値だが、人間が読む場合には 8 ビットずつに区切った 4 つの部分を１０進数（０～255）の組みとして表記

・インターネットに接続するホスト（通信機器）は必ず一意の IP アドレスを持つ（世界中でただ１つであることが原則、重複は許されない）

○ ネットワークアドレスとホスト番号

・**ネットワークアドレス**　：IP アドレスの上位ビット

・ネットワークごとのアドレス

・どの組織の所属かを分類できる

・現在地からの「近さ」もある程度判定できる

・**ホスト番号**　：IP アドレスの下位ビット

・ホスト（コンピュータ）ごとのアドレス

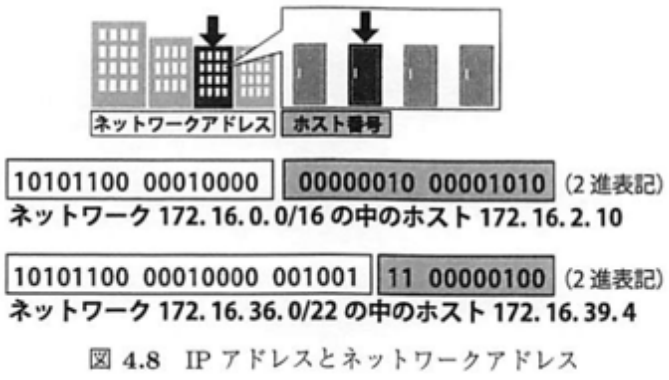
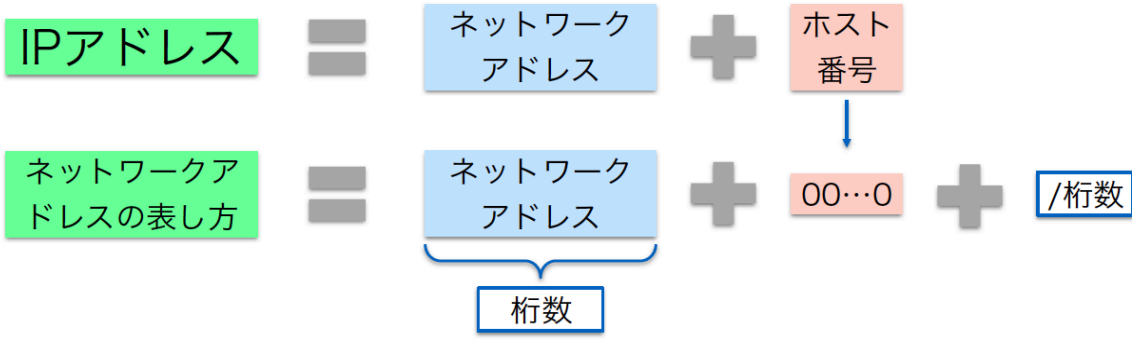


図 4.8　IP アドレスとネットワークアドレス

・どこまでがネットワークアドレスを表すのか？　→場合により異なる



［ネットワークアドレスの表し方］

・ホスト番号の部分を０とした IP アドレスに、斜線（/）に続けてネットワークアドレスのビット数を記してネットワークアドレスを表す。

例） **192.168.12.240**

＝11000000　101010000　00001100　11110000　（２進表記）

・192.168.12.241 192.168.12.254 までがそのネットワークで使えるホストのアドレス（192,168.12.255 は同報（＝同時に多数の相手に送信）のために使う）

・このようなネットワークを　**192.168.12.240/28**　と書く

・２８は先頭から 28 ビットまでがネットワークアドレスの範囲

○ ポート番号

・**ポート番号**　：同じホストの複数の通信を区別する番号

・機器の住所は IP アドレスで区別

・同じ機器内でも複数のアプリケーションが通信できるように

（ウェブブラウザとメールクライアント　・・それぞれが別のポート番号）

・16 ビットの数値

＜4－2－５　トランスポート層（TCP）とインターネット層（IP）＞

TCP　：仮想的な１対１で双方向の通信路を作成し、アプリケーション層のメッセージの送受信を実現する。下位の IP プロトコルを利用する。

（→参照　４－２－２）

①送信したいデータをパケットで送れるよう一定の大きさに**分割**

②荷札に相当する **TCP ヘッダ**をつける

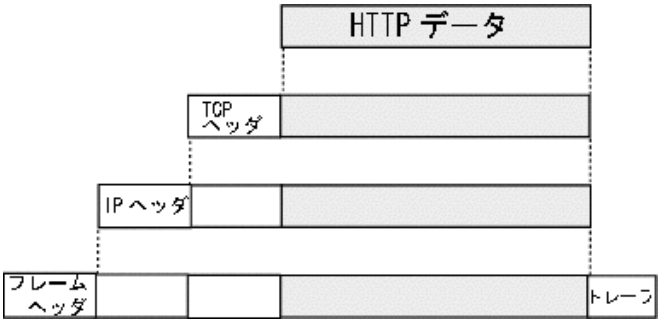
→ポート番号、シーケンス番号（分割した順序）、誤り検出のための符号

・このように適切なラベルを付与して下位の層に渡すことを**カプセル化**という

○ （参考）ヘッダとトレーラ

・**ヘッダ**：先頭に付加されたもの

・**トレーラ**：末尾に付加されたもの

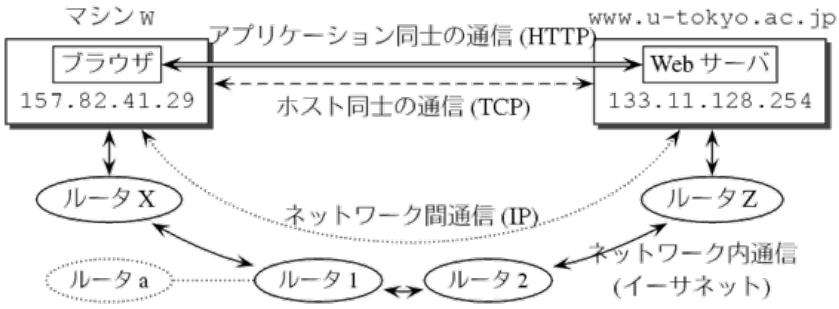


役割

・データの宛先

・誤り訂正

・順序の制御など



・受信側・・・届いたパケットをシーケンス番号を用いて正しい順序に並び替えて結合し、元のデータを復元

・IP パケット消失に備え受信側は確認応答（ACK）を返す

→送信元は必要に応じてデータを再送したり、

ネットワークの混雑状況を指定して送り出す速度を調整

・**UDP**　：TCP と同じトランスポート層のプロトコル

途中にデータが消失しても再送を行わない

・再送により全体が遅延するよりその部分を無視して最新のデータの受信を優先したい場合状況に

例）ストリーミング中継（リアルタイムで動画をフレームごとに配送）

IP　：パケット（データ片）を指定の IP アドレスに届ける

・IP パケット：IP プロトコルで配送されるデータ片

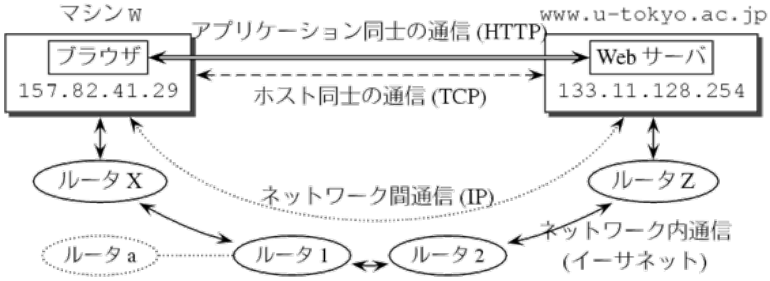
・ヘッダには宛先や送信元の IP アドレスなどが書かれる

・パケットの配送方法

・宛先のアドレスが管理組織の内部かどうかを判定

内部　→　適切な通信機器に配送

外部　→　ルータに配送



○ ネットワーク間のパケットの配送方法

・静的経路制御

各ルータに＜宛先ネットワーク、担当ルータ＞というペアを経路表に登録し、配送先の決定に用いる。

→ネットワークの数が多くない場合

宛先ネットワーク	担当ルータ
lecture.ecc.u-tokyo.ac.jp	ルータA
www.apple.com	ルータB
www.google.com	ルータC
・・・	・・・

経路表の例

・動的経路制御

近隣のルータと情報を交換しながら経路表を更新

ネットワークプロバイダなどインターネットと複数の出口で相互に接続している場合

○ **防火壁（ファイアウオール）**

（図４－４を適宜見ながら想像して見てください）

・防火壁（ファイアウオール）

：組織とインターネットの接点におかれ、そこを通過する通信を制御・偽造の疑いのある不審なパケットの廃棄を行う

例）　TCP や IP のヘッダを検査して、宛先や送信元の IP アドレスやポート番号に応じてパケットを選択に通過させる

→　組織内からは外にアクセスして返信を受け取る
組織外からは特定のサーバのみにしかアクセスできなくする

§ 3　通信の秘密と相手の認証

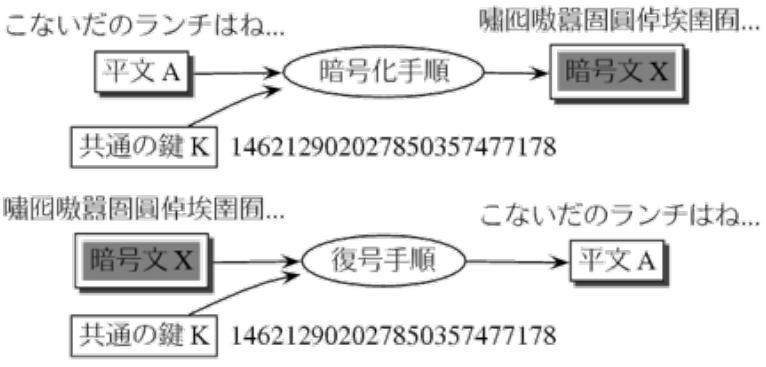
＜ 4-3-1　共通鍵暗号と公開鍵暗号＞

・**平文**　：元のデータ。第三者に読まれたくないもの。
例）「明日のランチはね・・・」
・**暗号文**：暗号化したデータ。盗聴されても平文を（簡単には）取り出せない。
例）「嘯囮噭囂圄圓倬埃圉圃…」
・**暗号化**：平文から暗号文を作成すること。計算手順と鍵を用いて行う。
・**復号**　：暗号文から平文を取り出すこと
・**鍵**　　：暗号化や復号の際に用いられるデータ

例）ある文章を暗号化する手順が「五十音を１文字ずらして置き換える」の時
→「ずらすという操作」が計算手順、「ずらす文字数＝１」が鍵となる

○ 共通鍵暗号と公開鍵暗号

・**共通鍵暗号**　：暗号化と復号に共通の鍵を使う方式
・初代「標準暗号」の DES やその後継の AES など
・鍵を秘密に保つ必要がある（どのように相手と共有するか）
・鍵を相手に送るときに鍵を盗聴されたら秘密は漏れる。



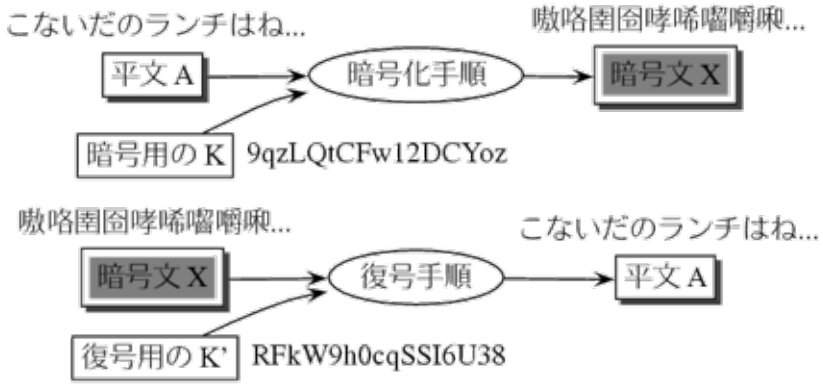
・**公開鍵暗号**　：暗号化の鍵と復号の鍵が分けられている方式

・２つの鍵（**公開鍵**と**秘密鍵**）の特徴

・暗号化の鍵（公開鍵）が異なれば複合の鍵（秘密鍵）も異なる。

・暗号化に用いた鍵では復号できない

・複合もための鍵を暗号化の鍵から推測することも難しい



・暗号化の鍵を公開し（＝公開鍵）、復号化の鍵を秘密にする（＝秘密鍵）

・手順

・各自が暗号化と復号のための鍵のペアを作成

・暗号化用の鍵を公開（公開鍵）

・復号用の鍵を自分だけの秘密に保持（秘密鍵）

→公開鍵を知っている人は誰でも暗号文を作成できるが、それを復号できるのは秘密鍵の所有者（受信者）だけ

・公開鍵暗号の例・・・RSA や楕円曲線暗号など

[共通鍵と公開鍵の組み合わせ]

・暗号化するための計算時間

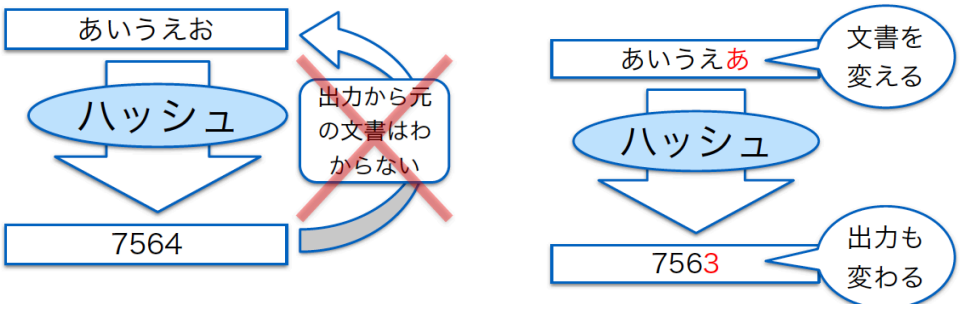
公開鍵暗号　＞　共通鍵暗号

→　両者を組み合わせて、共通鍵暗号の鍵を公開鍵暗号で安全に共有し、その後はその鍵を使った共通鍵暗号でデータを送受信することもある。

＜ 4-3-2　ディジタル署名と PKI ＞

・一方向ハッシュ関数：文書（電子的データ）を入力してそれを特徴づける数値を計算する関数

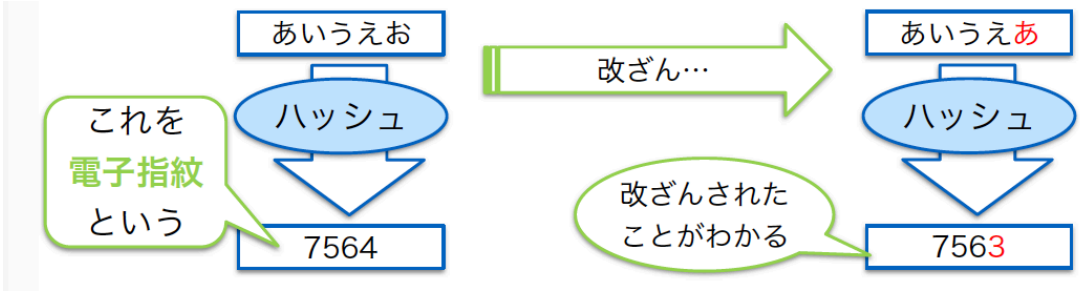
→　文書の改ざんの検知に役立つ。



・**電子指紋**（fingerprint）

：一方的ハッシュ関数の計算結果を文書に対する**電子的な指紋**とする

→　電子指紋が変わっていれば文書が改ざんされたことが検知できる。



・**ディジタル署名**

：電子的な署名の技術。「**誰が**」「**どのように**」署名したかについて第三者の検証を可能とする。

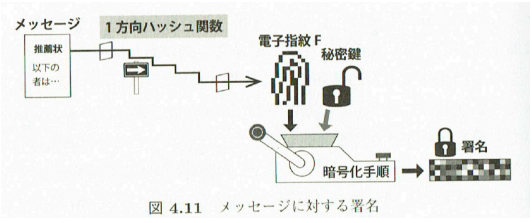


図 4.11　メッセージに対する署名

・実現方法

・秘密鍵（署名者しか所持しないはず）を利用

署名者の公開鍵（受信者が持つ）で検証する

・（RSA 暗号では）通常の手順とは鍵と暗号化/復号の対応を変えて、秘密鍵で暗号化を、公開鍵で復号することもできる（復号は誰でもできるが暗号化できるのは本人だけ）ことを利用　　（→暗号化の手順とは逆！）

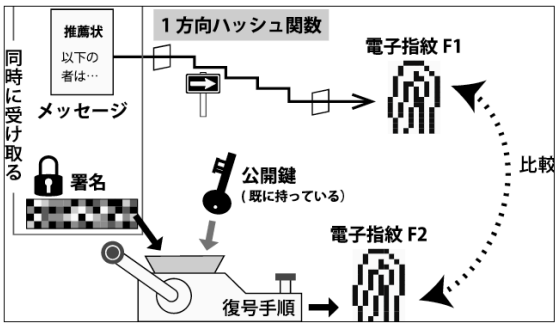
［実現方法を詳しく見ていこう］

・送信者が作成した文書の電子指紋を送信者の秘密鍵を用いて暗号化して、これをディジタル署名とする。

・受信者は文書本体と署名の双方を受信

・文書全体から一方向ハッシュ関数で計算した電子指紋

・（送信者の鍵を用いて）署名から複合した電子指紋



●この両者が一致！

→秘密鍵の所有者が作成したもの　かつ　改ざんされていない

（※　秘密鍵が漏れてしまうと署名が意味をなさないことに注意）

○ 公開鍵への署名

・公開鍵暗号を用いて秘密文書を送るとき

所持する公開鍵が偽物だと？　→　意図しない相手に複合される

偽の署名を信じる

・公開鍵を受け渡す際に第三者が公開鍵にディジタル署名を行う

署名を検証→その公開鍵が**署名時点から**改ざんされていない

署名者がその鍵を正しいと**信じている**

・・・その第三者が信頼できるなら公開鍵も信頼できるということになる。

（**PGP**（Pretty Good Privacy）という電子メールで使われる暗号化と認証の仕組みでは信頼の輪というモデルを用いている）

○ ディジタル証明書

・HTTPS でウェブサーバを利用者が認証する場合

・ウェブサイトは自身が本物であることを示すために**ディジタル証明書**を提示

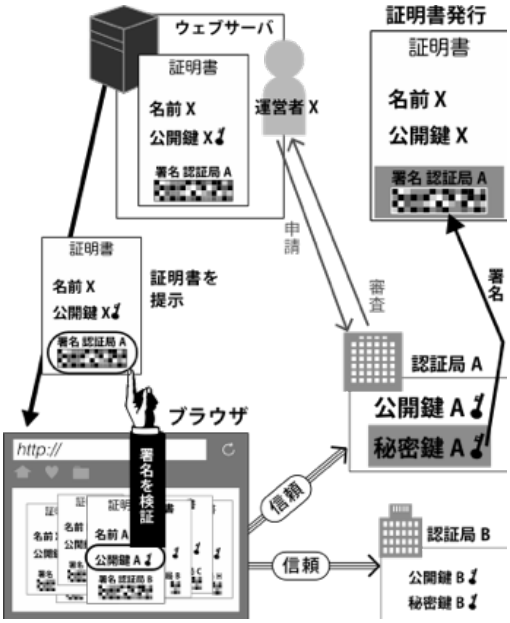
・ウェブブラウザがディジタル署名の正しさを検証

・**ディジタル証明書**：ウェブサイトの公開鍵と組織名、証明書を発行した認証局によるディジタル署名などの情報

○ 認証局と PKI

・ルート証明書

：ウェブブラウザにはあらかじめいくつか認証局のディジタル証明書を信頼するものとして登録しておくもの



・知らない認証局の署名

→その認証局を認証した親の認証局を検証

→さらにその親の認証局・・・(繰り返し)

→最終的にルート証明書へたどり着く

→最初の証明書も信頼する

PKI (公開鍵暗号基盤)

：認証局を通じて公開鍵の正当性を信頼するこのモデル（仕組み）

第５章　　「計算の方法」

§１　計算とその記述方法

計算とは・・・モデル化されたデータに対する種々の操作

＜5-1-1 計算の方法＞

・**計数**（counting）：計算の一種

「ある集合Ａの要素数を求める」という計数を例に考える。

（a）取り出し型　　・・・１つ１つ指折り数えるやり方

〔用意されている処理〕

・空（要素が１つもない）かどうか判定する

・要素を１つ取り出す（集合の要素は１だけ減る）

〔計算〕

・＜答＞を０にする

・Ａが空でない間、以下の処理を繰り返す

　　要素を１つ取り出す

　　＜答＞を１増やす

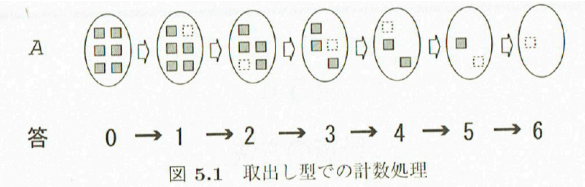


図 5.1　取出し型での計数処理

（b）分割型　　・・・手に余る仕事は下請けに出す

〔用意されている処理〕

・空あるいは要素が１つだけであるかどうか判定する

・空でない２つの集合に分割する

〔計算〕

・Ａが空なら＜答＞は０、

　要素数が１なら＜答＞は１

・そうでなければ

　　Ａを Ｂ と Ｃ に分割

　　（B、Ｃも空集合ではない）

　　＜答＞＝Ｂの要素数＋Ｃの要素数

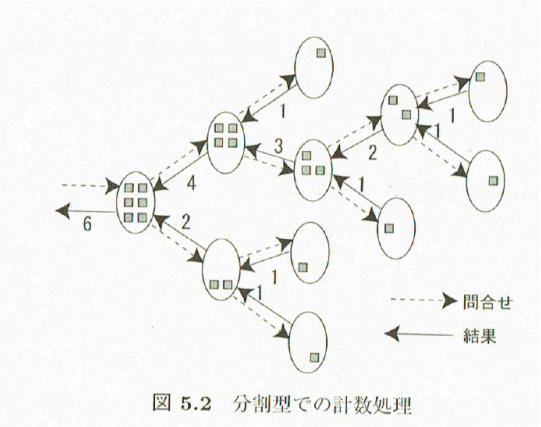


図 5.2　分割型での計数処理

＜5-1-2　計算の記述＞

（a）変数と条件判断

〔問題〕　今年の八十八夜は何月何日か。ただし今年の立春は２月４日であり、今年は平年である。

八十八夜は立春から８８日目（８７日後）である。

・２月４日の８７日後は２月を越す→２月の残り日数（28－４＝２４日）を引く

８７日－２４日＝６３日

・３月６３日は３月を越す　→３１日を引く（６３－３１＝３２日）

・４月３２日は４月を越す　→３０日を引く（32－３０＝２日）

・５月２日は５月に収まる！→　最終的な答えは５月２日

これをもう少し詳しい（＝曖昧さのない）レベルで書き下してみる

・２月４日の８７日後を求めたい

→２月９１（＝４＋８７）日という仮想日付になる

・＜残り日数＞を９１にする

９１＞２８（２月の日数）なので

＜残り日数＞から２月の日数を引き、３月に進む

→３月６３（＝91－２８）日という仮想日付になる

６３＞３１（３月の日数）なので

＜残り日数＞から３月の日数を引き、４月に進む

→４月３２（＝63－３１）日という仮想日付になる

３２＞３０（４月の日数）なので

＜残り日数＞から４月の日数を引き、５月に進む

→５月２（＝32－３０）日という仮想日付になる

２＜３１（５月の日数）なので、計算終了。

○**変数（variable）**

先ほどの計算手順で＜残り日数＞は初め９１に設定されるが、その後だんだんと小さい値に変化していく。このようなものを**変数（variable）**という。

・**変数**　：値を覚えておくもの（今回の例では値＝残り日数）。
・**変数名**：　“残り日数”　という文字列自体のこと。
・**代入**（assignment）　：変数に値を代入する操作
　→変数の値は「代入」により様々に変化させることができる。

　・代入の操作は　　「変数名」　←　「式」　と表記する。

○ 計算の手順

・**逐次処理**（sequential processing）
　　：書かれた順序通りに処理すること。途中を抜いたり、２つのことを同時にやってはいけない。
・**条件付き処理**（conditional processing）
　　：条件によって実施すべき操作が異なり、操作を切り替えていくこと。

[条件付き処理の表記方法]

```
If 条件
  then    条件が成立した場合に行う処理
  else     条件が成立しない場合に行う処理
endif
```

ただし、else 以降がない場合には、else も含めて省略する。
実際に先ほどの問題をこの表記方法で計算してみる。

```
＜残り日数＞　←　４＋８７
if ＜残り日数＞　＞28（2 月の日数）
   then＜残り日数＞　←　＜残り日数＞ー2 月の日数
      if ＜残り日数＞　＞31（3 月の日数）
         then＜残り日数＞　←　＜残り日数＞ー3 月の日数
            if ＜残り日数＞　＞３０（４月の日数）
               then＜残り日数＞　←　＜残り日数＞ー４月の日数
                  if ＜残り日数＞　＞３１（５月の日数）
                     then　　（６月以降の処理）
                     else　　“5 月”＜残り日数＞“日“と表示
                  endif
               else　“４月”＜残り日数＞“日”と表示
               endif
         else　“３月”＜残り日数＞“日”と表示
      endif
   else　“２月”＜残り日数＞“日”と表示
endif
```

・endif とそれに対応する if　は同じ欄に縦に並ぶように書かれている。
・then と else　の関係も同様である。
これは「まとまりの構造」をわかりやすく表すためである。
　（このやり方を字下げ（indentation）という）

[この手順の問題点・改良点]
・目的の日付が５月までで収まる場合しか書かれておらず、６月以降については「(６月以降の処理)」として曖昧に省略されている。
　→実際６月以降の処理が必要になった時、正しい答えが求まる保証なし。
・もっと簡素化できるはず
・「２８（２月の日数)」という形が何度も現れている。

(b)　繰り返しと添字つき変数

○　繰り返し実行の記法

```
while　条件　do
    繰り返し実行する処理
done
```

・「ある条件が成立している限り、指定された処理を繰り返して実行する」という指示を表す。

・このような処理を**反復処理**（repetitive processing）という。

● 先ほどの「２８（２月の日数)」を $daymonth_2 = 28$ と書けることにする。

・[実際の処理手順]

```
＜残り日数＞　　←　４＋８７
m　←　２
while ＜残り日数＞　＞　daymonth_m　do
        ＜残り日数＞　←　＜残り日数＞－ daymonth_m
        m　←　m＋１
done
```

・**配列**（array）：複数の値を一列に並べたもの

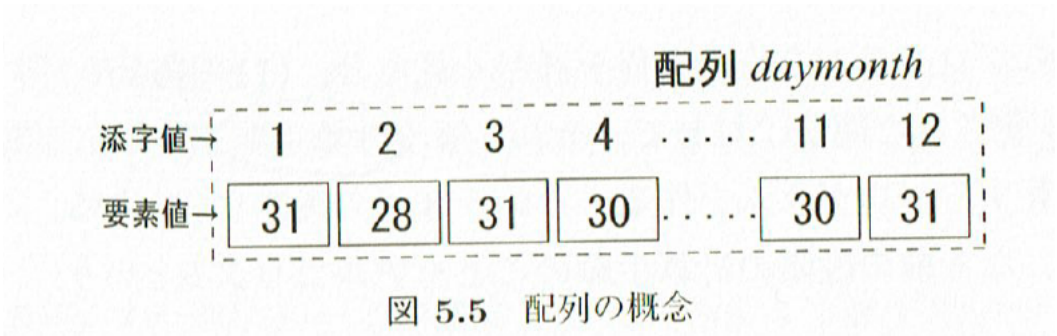


図 5.5　配列の概念

§２　　アルゴリズム

・コンピュータによって問題を解く＝人間が書いたプログラムをコンピュータが実行

・問題からプログラムを作る過程

（１）問題をモデル化する

（２）モデル化された問題に対してそれを解く計算手順を考える

（３）手順通り計算するプログラムを作る

アルゴリズム　：（２）における計算の手順のこと

＜5－２－１　アルゴリズムの実例１：平方根の計算＞

（目的）一つの問題に対して複数のアルゴリズムがあり、アルゴリズムにより答えを求めるまでの回数が異なるような例を見る。

〔問題〕

・ある正の実数 x が与えられた時に、２乗すると x に近くなる正の実数 y を、精度 δ で求める。

・すなわち、$\left|\sqrt{x}-y\right|<\delta$　となるような y を１つ求める。

● アルゴリズム１　・・・反復による平方根の計算

y ← 0	初めに y は 0 に設定（y に 0 を代入）
while $(y+\delta)^2 < x$ do	$(y+\delta)^2 < x$が成り立つ限り以下を行い繰り返す
y ← $y+\delta$	y に $y+\delta$ の値を代入し、再び$(y+\delta)^2 < x$を計算
done	$(y+\delta)^2 < x$　が成り立たなくなったら終了
return y	終了した時点の y を解として出力

・ **return y** は y を解として計算を終了するという意味。

例）　x＝９０、δ＝１　の場合、アルゴリズムに値を入れて見ると
y＝0,1,2,3・・・を順に検討してゆき、$(y+\delta)^2$が９０より大きくなったらその一つ前のｙが答えとなる。

・実際に反復法によって２の平方根を精度δ＝０.0001 で求めると、ｙ と $(y+\delta)^2$は以下のように変化していく。

回数	0	1	2	・・・	14140	14141	14142
候補 (y)	0.0000	0.0001	0.0002	・・・	1.4140	1.4141	1.4142
$(y+\delta)^2$	0.00000	0.00000	0.00000	・・・	1.99968	1.99996	2.00024

しかし、この方法は例えば精度が１桁変わると、計算回数が１０倍になる。
ではより効率的な方法はないのであろうか？
→解の存在する範囲に注目し、それを狭めていくという方法
　これを単純化したものが、以下に掲載する**二分法**である。

● **アルゴリズム２**　　・・・二分法による平方根の計算
・xの平方根を精度δで求めるが、ここでは　x＞１という条件設定をしておく。

a ← 0	まずａを０と設定
b ← x	ｂをｘと設定
while b－a＞δ **do**	b－a ＞δが成り立つ限り、以下を行い繰り返す
c ← $\frac{a+b}{2}$	ｃを$\frac{a+b}{2}$と設定する
if c^2 ＞ x	c^2 ＞ xが成り立てば、今回のｃを次のｂにする
then b ← c	成り立たなかったら、今回のｃを次のａにする
else a ← c	
endif	
done	b－a ＞δ　が成り立たなくなったら終了
return a	終了した時点でのａを解として出力

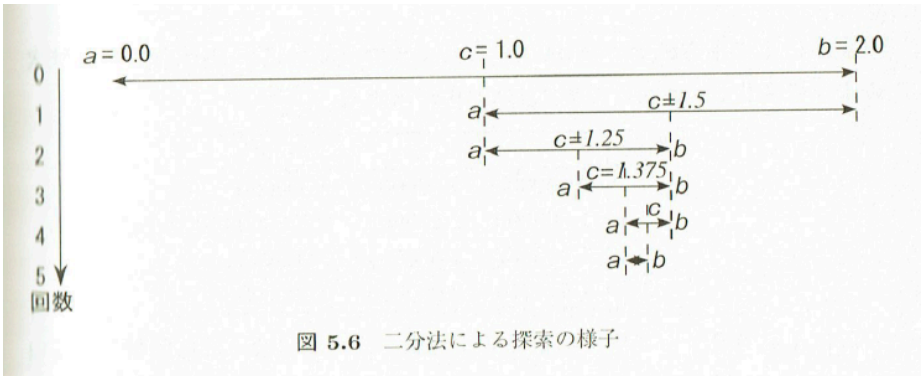


図 5.6　二分法による探索の様子

例）　x＝２、δ＝0.0001　の時

回数	a	b	区間の幅	c	c^2
0	0.000000	2.000000	2.000000	1.000000	1.000000
1	1.000000	2.000000	1.000000	1.500000	2.250000
2	1.000000	1.500000	0.500000	1.250000	1.562500
3	1.250000	1.500000	0.250000	1.375000	1.890625
4	1.375000	1.500000	0.125000	1.437500	2.066406
			（中略）		
13	1.414062	1.414307	0.000244	1.414185	1.999918
14	1.414185	1.414307	0.000122	1.414246	2.000091
15	1.414185	1.414246	0.000061		

（参考）●アルゴリズム３　　・・・ニュートン・ラフソン法による計算

y ← x

while $\frac{|x-y^2|}{2y} > \delta$ do

y ← $\frac{y^2+x}{2y}$

done

return y

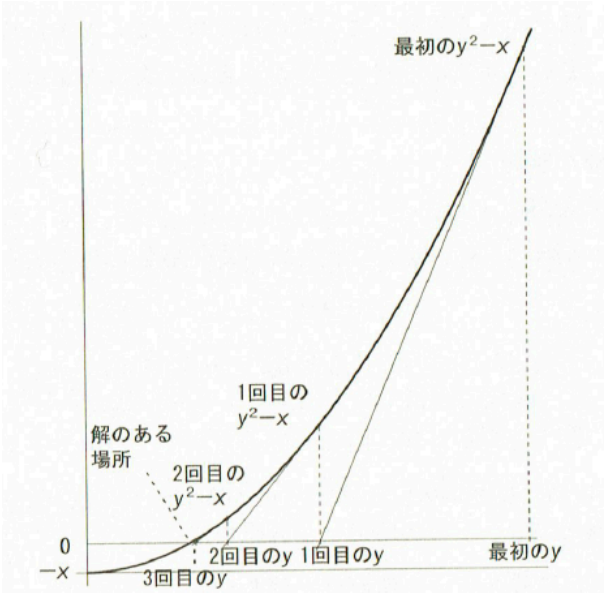


図 5.7　ニュートン法による平方根の計算の様子

回数	y	y^2	$\lvert x-y^2\rvert$
0	2.000000	4.000000	2.000000
1	1.500000	2.250000	0.250000
2	1.416667	2.006944	0.006944
3	1.414216	2.000006	0.000006

・これら３つのアルゴリズムで２の平方根を求めた場合、計算回数が14142回、１５回、３回と大きく異なる。いかにアルゴリズムの選択が重要かわかるだろう

＜5－2－4　計算量＞

(a) 計算量の考え方

計算量　：アルゴリズムをもとにしたプログラムの実行時間を見積もるための指標。**計算量のオーダー**という大まかな尺度で考える。

例えば、N 個のデータを処理する問題のアルゴリズムが２つあるとする。

・１つは一秒の処理を N 回繰り返すので、答えを出す時間は N 秒

・もう一つは、10 秒の処理をlog_2N回繰り返すので、時間は１０log_2N 秒

→　計算量のオーダーを考える際は、

前者は「N に比例」に比例する時間が、後者は「log N に比例する」時間がそれぞれかかるという点に着目し、他の違いは無視する。

（b）計算量の例

先ほどの問題に対する、２つのアルゴリズムの計算量を x、δをもとに考える。

・アルゴリズム１では$\sqrt{x}/\delta$回の繰り返しが行われる。

→計算量のオーダーは$\sqrt{x}/\delta$回となる。

・アルゴリズム２では最初の区間の幅が x であり、１回の繰り返しで幅が半分となり、幅がδより小さくなれば終了するので、繰り返しは$x/2^n < \delta$

→計算量のオーダーは log(x/δ)

第７章　　「データの扱い」

§１　データモデル

＜7-1-1　データとデータモデル＞

・**データ**　：（この章では）**コンピュータの処理対象となる符号化された情報** ・**データモデル**　：データを体系的に扱うためのモデル

→大量のデータを扱うデータベースでは、データを体系的に整理し、かつデータの解釈にブレがないようにすることが重要。そのためにデータモデルは発展。

§２　代表的なデータモデルと演算

＜7－2－２　ネットワークモデル＞

まずは「ケーニヒスベルクの橋」を例にとってみる。

〔問題〕　「同じ橋を二度渡らずにして全ての橋を渡れるのか？」

→この問題は「４つの陸地が橋を通る道でどのように連結されているか」だけで決まることが分かる。

→まずそれぞれの陸地をノード、道をノード間のエッジで表す　→図７.6
（ノード、エッジに関しては第３章§１を参照のこと）

→さらに、元の川と地面を消したのが図 7.7(→グラフ化した)

図 7.5　ケーニヒスベルグの橋

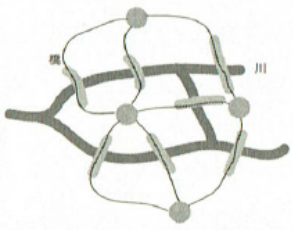


図 7.6　橋と道の抽象化１

図 7.7　橋と道の抽象化２（グラフ）

・**グラフ**　：ノードとノードをエッジで結んだデータ（図 7.7）
・**ネットワークモデル**
　：グラフなどのように「つながり方」を表すモデル一般のこと。

・**経路**　：順にたどっていけるエッジの列のこと
・**オイラー路**：全てのエッジを重複なくたどる経路

○ **ウェブ**

・ウェブのそれぞれのページをノードと考えてみる。
　→リンクはノードからノードを指す有向エッジと見ることができる。
　　ノードが有向エッジで結ばれたものもグラフとなる。（図 7-8）

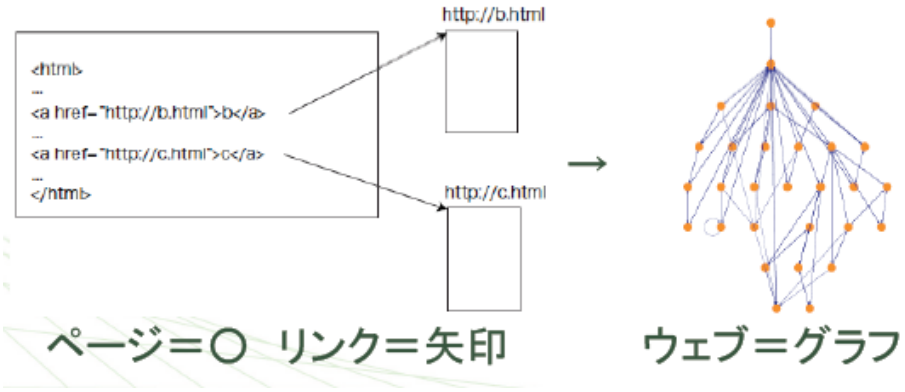


図 7－8

→「重要なページからリンクされているページは重要である」という規則の
　方程式を、このグラフを作って解くことで、各ページの重要度を決定できる！
→サーチエンジン（Google 等）ではキーワードによる検索結果を利用している
　この重要度の順に並べて目的のページを早く見つけられるようにしている。

＜7－2－3　階層モデル＞

・**木構造**：　枝分かれした構造

　　根：　枝分かれの分かれる大元のこと。

（他のシケプリからの参照：階層モデル/木構造について詳しく）

「ひとつ上位の要素に対して、ひとつ以上の要素が下位に存在し、ある要素はより上位の要素を用いて一意に特定できる性質をモデル」

・木構造は有向グラフの特殊なものと見ることもできるが、木構造では向きが自明なので矢印が省略されて線で描かれることが多い。

○ 階層的ファイルシステム

・・・コンピュータのファイルシステム
　　も木構造になっているものが多い

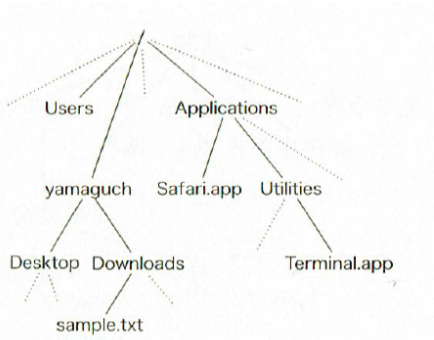


図 7.16　階層的ファイルシステム

[コンピュータのファイルシステムの特徴]

①フォルダの中にサブフォルダ
（木構造の中では**部分木**という）
　を作って細かい仕分けが可能

②サブフォルダが混合しない

③全てのフォルダとファイルを重複なく一度だけ処理できる

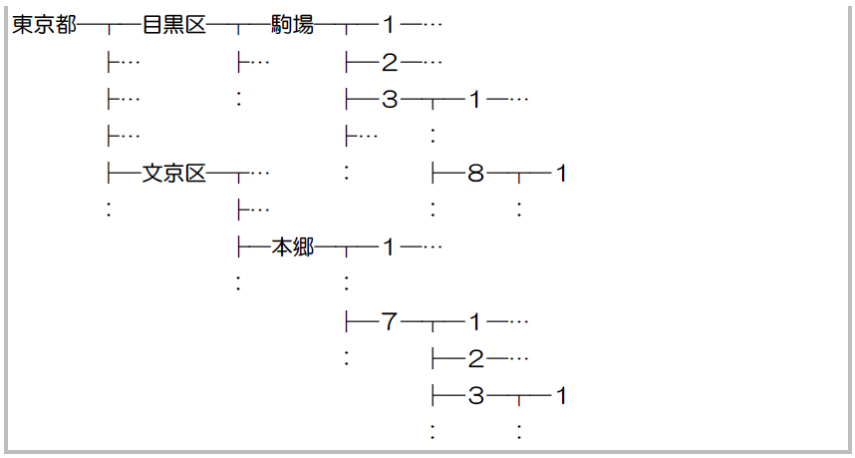
④サブフォルダ以下の内容をそのサブフォルダで代表させることが可能
　（例えば階層構造の中でサブフォルダ移動→一緒にその中身も移動）

○ 住所の階層性

　・・・住所も階層的な構造

・日本にある場所はどこでも「都道府県」から「市区町村」をたどって一意に住所が決まり、あいまいさが無いようになっている。

・「東京都港区」「名古屋市港区」「大阪市港区」のように、「港区」だけではわからないが、それに至るまでの公共団体名を並べると区別がつく。
　→こうした根からそこまでの名称まで並べたものを**パス名**という。



・また、コンピュータのドメイン名でも階層構造が使われている。

例）　u-tokyo.ac.jp の場合

jp(日本)

↓

ac (大学等)

↓

u-tokyo（東京大学）

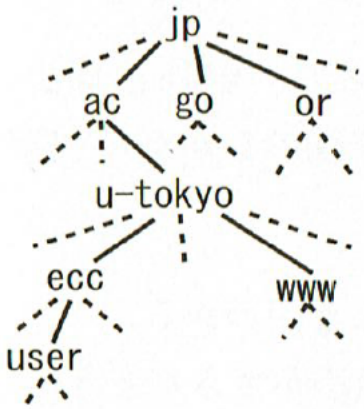


図 7.18　ドメイン名の階層構造

第８章　「コンピュータの仕組み」

§１　プログラム内蔵方式

・コンピュータの扱う情報は**プログラム**と**データ**の２種類に大別

プログラム	：コンピュータの実行する計算処理手順に関わる情報
データ	：コンピュータが処理する対象の情報

→これらの情報を総称して**ソフトウェア**（software）と呼ぶ。
また、ソフトウェア処理する物理的な機構を**ハードウェア**（hardware）という

・**プログラム内蔵方式**：メモリ上にプログラムとデータを保持し、プログラムに従って計算を進めるといったハードウェアの実現法
フォン・ノイマン型コンピュータ：プログラム内蔵方式のコンピュータのこと

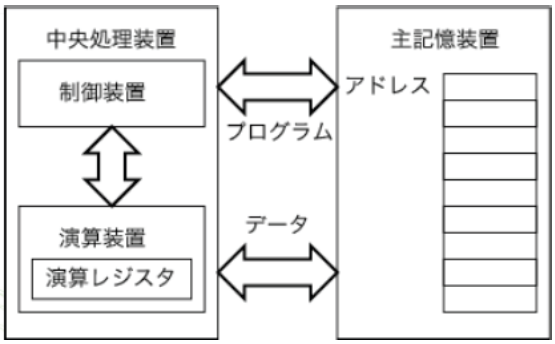
＜8－1－１　コンピュータの基本構成＞

○ プログラム内蔵方式の構成

・**演算装置**　：データに対して計算処理を施す装置
・**主記憶装置**（**メインメモリ**）：複数のデータやプログラムを記憶する装置
・**制御装置**
：主記憶装置上のプログラムに従って演算装置を駆動するとともに、主記憶装置へのデータの読み書きを行う部位

・**中央処理装置**
（CPU　：Central Processing Unit）
：制御装置・演算装置を組み合わせたもの

・**マイクロプロセッサ**
or　MPU(Micro Processing Unit)
：中央処理装置を１つの半導体集積回路として実現したもの



・**演算レジスタ**：これもデータを保持する装置。CPU における計算は基本的
にこの演算レジスタ内のデータを対象とする。
演算結果のデータも一旦ここに格納される。
・・・初期のコンピュータでは演算レジスタが 1 つだけしかなかった。
そのようなコンピュータは**アキュムレータ**（**AC: accumulator**）と呼ばれた

○ 主記憶装置（メインメモリ）
・・・複数の情報を格納し、選択的に情報を読み書きできる。
この際、情報の選択に**アドレス**（**address**）を用いる。
・**アドレス**（**address**）：主記憶上の位置を数字によって表す。主記憶装置上
の隣り合った場所には、連続したアドレスが割り当てられる。

第９章　「ユーザインタフェース　－人に優しいデザイン」

§２　インタフェースとは何か？

＜9－２－１　インタフェースの定義と機能＞

○ インタフェースの定義：２つの異なる存在の境界面

・水と油のように異なる物質の間に存在するもの

・コンピュータの複数のハードウェア間やソフトウェア間

・この章で扱うインタフェース

：コンピュータなど人工物とユーザ（人間）との間

→「ユーザインタフェース」「ヒューマンインタフェース」と呼ばれる

○ インタフェースの機能

・ユーザはインタフェースを通して人工物を操作する

→人工物が機能を最大限に発揮する為には、使いやすいインタフェースが必要

例）ドライバー（ねじ回し）のインタフェース　＝握り

→人間は握りを通してドライバーを操作する。ドライバーが機能を最大限に発揮するためには、使いやすい握りが必要

○ コンピュータのインタフェース

・先ほどのドライバーの例でいうと、人間の道具への働きかけは、そのまま対象（ネジ）への働きかけとなる。

（例：硬いネジには太いドライバ、細い・折れやすいネジは細いドライバを）

しかし、コンピュータの場合、

道具への働きかけが、対象への働きかけと等価であるとは限らない

例）エンターキーを押す

・目的は力の伝達ではなく、情報（意思決定）の伝達

・伝達対象は状況により変化（かな漢字変換、ミサイル発射命令など）

道具	ドライバ	コンピュータ
インタフェース	握り	キーボード、マウス、ウインドウシステム、…
道具への働きかけ	回す	キーを押す、マウスを動かす、…
対象への働きかけ	ネジを締める・緩める	メールを送る、変換を確定する、ウインドウを閉じる、入力を取り消す、…

↑道具への働きかけと対象への働きかけが１対１に対応しているか？

＜9－２－２　インタフェースの二重接面性＞

○ 第一接面（操作インタフェース）

・ユーザ（心理的世界）と人工物（道具・機械の世界）の間

・直接的

○ 第二接面（制御インタフェース）

・人工物と物理的タスク（仕事世界）の間

・間接的

→ユーザの目的は物理的なタスクの実行であるが、操作可能なのは第一接面

→コンピュータなど、高度な人工物には二重接面性が存在する。

○道具への働きかけと対象への働きかけが１対１出ないことはインタフェースの二重接面性により説明可能となる

・第一接面への入力が同じでも、道具の状態に応じて第二接面への働きかけが異なる

・このため限られた操作で多様な仕事が可能な、汎用性のある道具が実現可能

＜インタフェースの二重接面性と汎用性＞

○ 道具・物理世界から仕事世界への対応づけは多様

・ワープロソフトによる文書作成

・データベースソフトによるデータの管理

・数値計算プログラムによる建物の構造計算

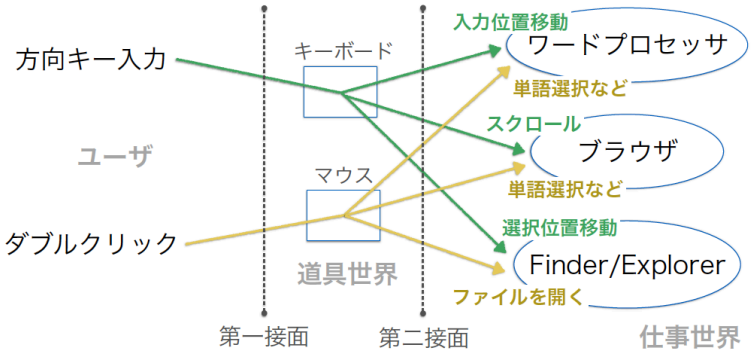
○ 心理的世界から道具・物理的世界への対応付けは限らている

・キーボード、ポインティング・デバイス、GUI(後述)

○ 逆向きの対応付けも同様

・コンピュータとソフトウェアの状態や実行結果は多様

・ユーザに提供される出力は限られている（文字、画像、・・）



§３　　実際のインタフェース

○ **入出力デバイス**

・入力デバイス（input device）

→コンピュータにデータや文字を入力する

・出力デバイス（output device）

→コンピュータの処理結果や状態を表示する

→ユーザの心理的世界とコンピュータの物理的世界との相互作用は、入出力デバイスとその上での表現を通して行われる

＜9－３－１　入力デバイス＞

・キーボード　　・・・QWERTY 配列が一般的（ユーザビリティに欠ける）

・ポインティング・デバイス

→２次元的な位置情報を入力するデバイス

（メニュー選択・アイコン操作に最適）

・直接入力型デバイス（ディスプレイ上で直接操作、直接位置指定）

例）ライトペンやタッチスクリーン

・間接入力型デバイス（ディスプレイ以外から操作）

例）デジタイザ、**マウス**、トラックボール、ジョイスティック

・その他・・・マイク、OCR 機器、バーコードリーダー

＜9－3－2　出力デバイス＞

・出力デバイスの例：**ディスプレイ、プリンタ、スピーカー**など

○ ディスプレイ

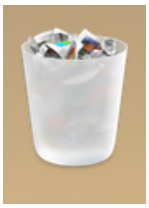
・かつては**ブラウン管**（CRT）が利用されていた→仕組み上薄型に限界が
→現在は**液晶ディスプレイ**（LCD）などの平板ディスプレイが主流

○ プリンタ

・**レーザプリンタ、インクジェットプリンタ**は高品質な印刷が可能であり、現在よく使われている

・**３Ｄプリンタの登場**（３次元立体を造形する機器）

＜9－3－3　GUI と CUI ＞



○ GUI (Graphical User Interface)

・情報の表示にウィンドウやアイコンなどのグラフィカルなオブジェクトを多用

・ポインティング・デバイスでオブジェクトを操作することで基本操作の多くを実現する

○ CUI (Character User Interface)

・情報の表示、入力を文字によって行う

・すべての操作をキーボードで行う

○ GUI と CUI の比較

	GUI	CUI
操作状況の表示	絵や画像も用いて視覚的に情報を表示する	キーボードから命令を文字で入力、文字列で結果を出力する
特徴	直感的で、わかりやすい	慣れれば、作業は迅速
構成要素と操作	ウィンドウ、アイコン、メニュー、マウスなどを用いた直接操作	プロンプトに続いて、文字列による命令を与える

○ GUI の歴史

・構想は早くは 1940 年代に、遅くとも 1960 年代には開始
・1960 年代に実用化開始
・アラン・ケイによるダイナブック構想（1968 年）と Alto の開発
・マッキントッシュへの採用（Apple 社）
・Windows, X window system (UNIX 系)の開発

○ **GUI の特徴**

・**デスクトップメタファ**　：机上に書類を広げる感覚での操作
・**直接操作の考え方**　：アフォーダンスの概念（後述）を拡張
　→その装置あるいは表示を見れば、どのように実行可能か即座にわかる
・**WIMP システム**
　：ウィンドウ（W)、アイコン（I)、メニュー(M)、（マウス）ポインタ(P)
　　を主な要素として構成（４つの頭文字を取っている）
　　・(マルチ）ウィンドウシステムにより実現

※アフォーダンスの概念
　：下界の環境や事物が、生体の活動に供するべく持っている情報
　　→例）椅子の形状は人が座るという情報を持っているとされる
今回の GUI の例でいうと、ゴミ箱マークはゴミを捨てる行為の情報を持つ

○ インタフェースの２つの側面
・物理的側面
　→・入出力デバイスの物理的特性とユーザの感覚運動系の特性の適合
　　・キーボードでは、押した時の感触やキーストローク
・認知的側面
　→・ディスプレイに表示されるメニュー項目の構成やメッセージの内容
　　・ユーザが内容を理解や記憶しやすいかが問題

§５　インタフェースの評価

・インタフェースが使いにくいとどうなるか？
　→ユーザ離れ、市場での競争力ダウン、顧客対応増加・・・
・インタフェースの良し悪しを評価する必要性
　→「良い」インタフェースを設計するために、まずは実証的に性能を評価することが重要
・インタフェースの実証的な評価：**ユーザビリティテスト**
　→パフォーマンステスト、ガイドライン法、モデル法、インスペクション法・・・

○ユーザビリティテスト

・パフォーマンステスト
　：システムの動作情報とユーザの操作履歴から評価
・ガイドライン法
　：既定の使いやすさのガイドラインを満たしているかどうかを評価
・モデル法
　：ユーザがシステムを利用する際の行動を行動モデルに即して評価
・インスペクション法
　：複数の専門家による画面例やモックアップ（模型、見本）から使い方を想定して問題点を見つける

・良いインタフェースデザインの原則
　1.可視性
　2.良い概念モデル
　3. 良い対応付け
　4.フィードバック

＜9－５－１　キーストローク・レベル・モデル＞

○モデル法の代表である **GOMS** の一手法

→**GOMS** ：Goal ,Operator ,Method,　Selection Rules

○ GOMS-KLM(キーストローク・レベル・モデル)

→特定の作業にかかる時間の予測値でインタフェースのユーザビリティを定量化するモデル

・作業をコンピュータの基本操作に分解し、各基本操作にかかる時間の目安を足し合わせる

・キーを押して離す（キーストローク）、クリック、マウスポインタの移動・・

○ キーストローク・レベル・モデルの計算例

[デスクトップ上のファイルをゴミ箱に入れる]

・ファイルをアイコンまでマウスポインタ移動（P）

→マウスのボタンを押す（B）

→ゴミ箱のアイコンまで移動（P）

→ボタンを離す（B）

→元の位置まで移動（P）

教科書表9.2：各操作に要する時間（秒）		
K	キーボードの1つのキーを押して離す	0.28 s
T(n)	キーボードからn文字入力する	n × K s
P	画面上の目標物までマウスポインタを移動する	1.1 s
B	マウスのボタンを押す、あるいは離す	0.1 s
H	キーボードからマウスに手を移動する（あるいはその逆）	0.4 s
M	心理的な判断もしくは知覚	1.2 s
W(t)	システムの応答時間	t s

・上の表を使うと、予測時間はＰ＋Ｂ＋Ｐ＋Ｂ＋Ｐ＝３.５秒

＜9－5－2　フィッツの法則＞

・人がポインタを始点から目標物まで動かすのにかかる時間についての法則

$$T = a + b \cdot log_2(\frac{D}{W} + 1)$$

・T：平均時間、a：ポインタの移動開始/停止にかかる時間
　b：ポインタの移動速度、D：始点から対象物の中心までの距離
　W：対象物の大きさ

・a はデバイスごと、b はデバイスとユーザの習熟度ごとに変化する

・要は遠くで小さい目標物ほど時間がかかる
　→ポインティングデバイスの評価に利用される

第１０章　　「情報社会と技術」

§1　技術と社会

○ 情報技術の発展は社会のあり方を大きく変えつつある
・通信技術と GPS の併用
　→　船舶・自動車のナビゲーション /防犯 / 軍事のあり方に影響を
・ビッグデータ
　→　新たなビジネスの対象として注目される
→**情報リテラシー**の必要性が高まっている

・一般にリテラシーとは、読み書き能力、識字率のこと
　・科学技術リテラシーとは
　→科学や技術を使う上での基本的な能力、科学・数学・技術に関係した知識・技術・物の見方。物事を論理的に考える能力も含まれる。

○ 情報リテラシーとは
・ただ単にパソコンを操作できるという意味の情報機器操作能力ではない。
　→・情報を主体的に選択、収集、活用、編集、発信する能力を持つこと
　　・情報機器を使って論理的に考える能力
・マウスでクリックした時に、「裏で何が動いているのか」についてのおおまかな想像力が及ぶ程度の理解能力

○　情報技術の普及
・良い側面と悪い側面の双方が同居。
・場面場面で技術開発者、利用者、法律の専門家や倫理の専門家などの相互の議論が必要。
○　<u>我々の今の選択は、将来世代の情報技術に影響を与える</u>
・「文明時代の野蛮人」になるのではない
・問題が反省するごとにそれらの議論に「参加」し、新しい社会規範の構築に参加できる情報リテラシーを身につけるべきである

§２　　情報技術の影響

(以下は序論です。時間がない人は軽く読み飛ばしても OK)

現代社会において科学研究及び技術開発の成果は、社会全体やその構成員の将来を左右するような形であらわれる。

[ライフサイエンスと医療]
・新しい治療法や、生殖医療、再生医療の応用は、社会の構成員の一人一人の生や死と直結
[情報技術]
・技術の流通は、社会構成員一人一人のリスクや安全、セキュリティとプライバシーの問題と直結。

・科学技術に関係した（安全と安心に関わる）重大な社会的政治的問題が発生した時、未来を選択する権利は、民主主義社会においては国民一人一人にある。
→社会の構成員は、科学技術の研究成果について、その選択に必要な程度の基礎知識を持っておく必要がある。

・情報についても然り
民主主義社会において、「情報」に関する選択主体は私達。
その選択に必要な程度の基礎知識は持っておく必要あり。
＝情報を学ぶ意義。

○　情報技術とどう付き合ってゆくべきか？　ーーー答えはない
○　情報技術が世の中にどのような影響を与えるのか？
　・良い・悪い影響両方ある
　・情報技術の特徴がどう関係するのか？
　・どのような点が問題になるか？

（さて序論はここまで。ここからは本題に）

＜10－2－1　技術上の変化＞

年代	技術	影響
'60s, '70s	商用・科学技術用コンピュータ	大規模なデータを処理が可能に "Big brother"
'80s	マイクロ コンピュータと ビジネスソフト・ゲーム	個人の道具としての コンピュータ・ソフトウェア ソフトウェアの所有権
'90s-	インターネット・WWW	個人による情報発信が可能に プライバシー、情報の所有権

○　それまで：事務処理は紙と手作業であった
・例: 国勢調査・戸籍管理・犯罪記録・預金管理
・データの保管場所 = 使用する場所 → 分散
・調べるのも手作業
○　コンピュータ登場後（'60s-）
・データを集中管理・集中処理（← ネットワーク）
・広範囲にわたる統計・検索が可能に
・広域サービス
→ 監視社会への懸念

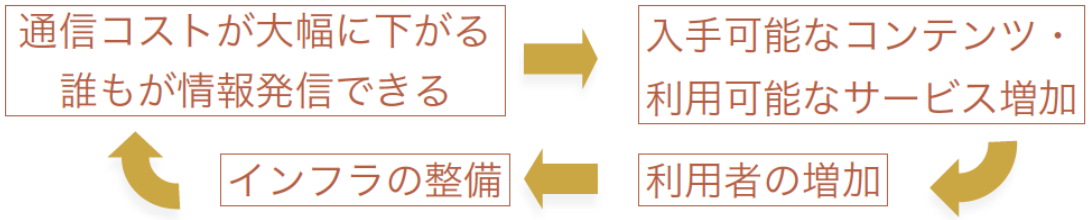
○　それまで：
・個人レベルの活動は道具+人間
・事務: 紙・ペン、ゲーム: モノ
○　マイクロコンピュータの登場後('80s-)
コンピュータ+ソフトウェアによって
・個人レベルの活動が支えられる
・　しかもソフトウェアが本質的
→ ソフトウェアの複製は容易（cf. 道具はコピーできなかった）

○それまで：

・通信にはコストがかかる

・放送は限られた人だけに許される

○　インターネット・WWW の普及（'90s～）



→　権力構造の変化、情報所有権の問題

○　インターネット技術

・ホストコンピュータ管理による閉じられた世界
（封建主義的）

・村（ネットワーク）中心の開かれた世界へ

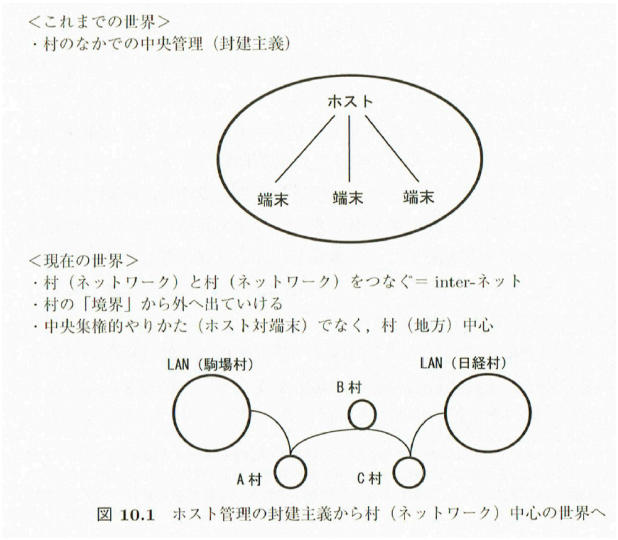


図 10.1　ホスト管理の封建主義から村（ネットワーク）中心の世界へ

○インターネットで村から村へ

・村の「境界」から外へ出ていける

・中央集権的なやり方（ホスト対端末）ではなく、村（地方）中心

○　情報技術の発展による４つの変化

１）場所の制約からの解放

２）時間の制約からの解放

３）経路の制約からの解放

４）輸送コストからの解放

・例）手紙や FAX によるコミュニケーションとメールなど IT によるコミュニケーション

この４つの変化はコミュニケーション形態を一変させた。

（以前）
・場所、時間、経路の制約
・国家や地域共同体のような地理的要素に依存したコミュニケーションが主流
（現在）
・制約からの解放
→地理的要素に依存しないコミュニケーションが可能に
→結果として国家や地域共同体のような地理的要素に依存した社会を相対化

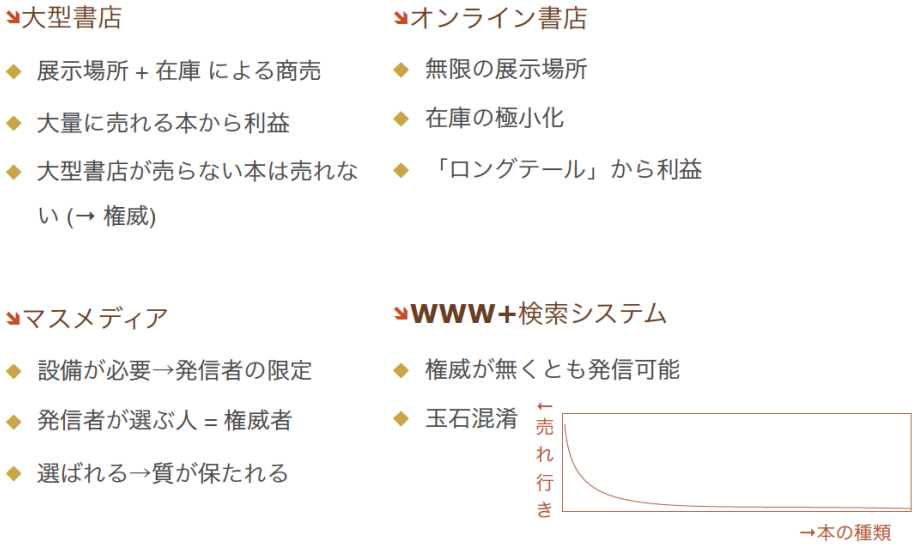
○　権威構造の変化

例１）　流通機構の変化と規制　（＝税制や法への影響）

○税務署も手の届かない巨大オークションの普及
　→近年（2013 年以降）では「フリマアプリ」が流行
　・流通経費がかからないため巨大店舗を持つものが優位を維持できない
　・従来の流通システムに対応した税システムが機能しなくなる可能性あり

例２）　系列会社優遇の壁の崩壊　（＝経済面への影響）

○ウェブ上の部品取引による系列会社壁の崩壊
　・瞬時にして最安値の部品を仕入れることが可能
　・系列会社優遇の障壁が少なくなる可能性あり



例３）　情報流通、検閲機能の低下　（＝情報検閲の問題）

〇検閲なしの情報の流通

・出版社などのチェック機能なしに、すぐに電子出版が可能

・個人のホームページを通じた情報の公開

・情報は玉石混交

例）原爆の作り方が個人で入手できる。個人の意見が「事実」のようにweb上に掲載される

例４）　文化や宗教への影響

〇イスラム教徒でもアクセスできるアダルト画像

・従来の検閲機能の機能不全、宗教への脅威

→・技術の普及によって、国境がない、法律が効かない、既存の権力が及ばない範囲での交流が可能になった

・しかも、国家はネットなしでは生きていけない

・アナーキー（既存の秩序が及ばない）でかつ必要不可欠、破壊と創造の同居

〇　技術領域を超えた問題の山積み

〇　既存の社会規範（法、倫理）によっては十分に制御しえず、新たな社会規範の形成が必要

・法の側面　：有害情報の規制、著作権の保護、知財の分類

・経済の側面：商取引の秩序と規制、流通機構の変化など

・文化の側面：情報検閲の可否、デジタルディバイド

〇　サイバースペースの良い影響

・情報技術を使った情報空間が公共的な議論を促進

・電子民主主義を促進。民主主義の発展に貢献する可能性。

＜１０－２－２　SNS、GPS、ビッグデータと社会の接点＞

○　SNS の発達

→議論の場がネット上にも展開

・炎上・サイバーカスケード現象

（※サイバーカスケード現象：短時間で同様の意見を持つもの同士が結びつけられ、結果として異質なものを排除する傾向を持つことにより集団極性化が生じる現象をいう。）

○　通信技術と GPS

→自動運転車の実現に寄与

○　ビッグデータ

→人工知能研究における深層学習の基礎

→IoT(Internet of Things)

：インターネットにコンピュータ機器というより通信機能付き組込みプロセッサとセンサーを取り込んだ機器（家電製品、ウェアラブル機器など）を接続するシステム

§３　社会への影響　　ー変わりゆく世界

・情報技術に特徴的な「社会との軋轢」とは何か？

○　コンピュータと情報の特徴

（１）無形性と複製可能性（ジョンソン、2002）

・無形性＝ソフトウェアやデジタルコンテンツが、将来の「モノ」概念とは異なる性質を持つこと

・所有と権利に関わる法制度や倫理との間の軋轢

（２）グローバルな通信射程と匿名性

・ネット上の自由で規制のない議論空間、公共空間の構築として、電子民主主義を促進や情報技術によるガバナンスといった展望を開く

・同時に、プライバシーやセキュリティといった側面での軋轢を生む

＜１０－３－１　権利と所有の境界　＞

〇情報技術は、市場に流通するものの「媒体」を変えつつある

例）これまでの CD、ビデオテープ、現像された写真、本、コピーといった物質的な媒体で扱われてきたもの

→　電子的に取り扱われることとなる（電子媒体）

↓

・保有あるいは所有といった概念も変容

複写する行為、著作権の侵害に対する利用者の意識の希薄さ、目の前のモノ（物質的媒体）の時は所有者がはっきりし、コピーを取ることに対して抵抗があったのに対し、電子媒体になった時はこのような抵抗が少なくなる。

→このことが端的に現れるのが、著作権をめぐる議論

（a）著作権法

（参考）[著作権法第１条]

この法律は、著作物並びに実演、レコード、放送及び有線放送に関し著作権の権利及びこれに隣接する権利を定め、これらの文化的所産の公正な利用に留意しつつ、著作者等の権利の保護を図り、もって文化の発展に寄与することを目的とする。

［著作権法により保護される著作物］

1. 小説、脚本、論文、講演その他の言語の著作物
2. 音楽の著作物
3. 舞踏または無言劇の著作物
4. 絵画、版画、彫刻その他の美術の著作物
5. 建築の著作物
6. 地図又は学術的な性質を有する図面、図表、模型その他の図形の著作物
7. 映画の著作物
8. 写真の著作物
9. プログラムの著作物

○　本・映画・音楽などを作った人を複製から守る権利
・創作の労力に比べて複製は非常に簡単
・複製技術が登場して以来の概念
○　文化の発展が目的
・作者が安心して製作できるようにする
・誰もが安心して二次利用できるようにする（パロディも文化）
○　「物質」に依らないものなので、問題点も多い
・過ぎたる保護は発展を阻害

（b）　コンピュータプログラムの著作権
○　1985 年の著作権法改正で「著作物」に
・保護されるのはプログラム
・プログラミング言語、アルゴリズムは対象外

著作物として保護される	ソースプログラム オブジェクトプログラム オペレーティングシステム（OS） アプリケーションプログラム
著作物として保護されない	プログラミング言語 規約 解法

事例）

○　A 社はビデオゲーム X を開発し、その著作権を所有

○　B 社は都内で経営する喫茶店にゲーム X の無断複製をビデオゲームとして設置して、顧客に利用させた。

○　A 社は著作権侵害を B 社に対し訴えることができるだろうか？

→答え　：　可能

この場合、ビデオゲーム機に取り付けられたＲＯＭに収納されているオブジェクトプログラムは、A 社の著作物（ソースプログラム）の複製物である。したがって、B が使用したビデオゲーム機のように、ＲＯＭのオブジェクトプログラムを他のＲＯＭにコピーして製造した偽造ゲーム機は、A のソースプログラムの著作権を侵害する。

（c）ディジタルコンテンツの著作権

○　デジィタルコンテンツ：音楽・映像・画像・電子書籍・ソフトウェアのファイルなどを指す

○　アナログコンテンツとの違い

・完全な複製が作れる

・簡単に複製が作れる

・（＋ネットワーク）低コストで広範囲に配布できる

・複製しないと利用できない場合もある

・システム次第では、利用方法を強く制限することもできる

（c.f.　デジタル放送のコピー制御）

[ネットワークによる情報発信]

○　放送と性質が似ている

○　放送との違い

・必要な設備が非常に安い

・許可がいらない

・「送信」ではない（受信側がサーバに要求すると、サーバが返答しているだけ）

・世界中に発信する場合も組織内で共有する場合も仕組みは同じ

→1997 年の著作権法改正で「送信可能化権」が「公衆送信権」の 1 つに

○　使用許諾の概念が曖昧

・メモリ上の電子情報は市場や流通機構なしに直接個人の手に渡せる

→　**○ Winny 開発者の起訴問題**

・Winny ：ファイル交換が完全に匿名化されていることが特徴

・ディジタルコンテンツの著作権の侵害を幇助する側面を持つ

○　Winny ：匿名ファイル交換システム

・中央集権的なサーバが不要

・端末同士でファイルを交換

・匿名：最初に公開した端末がわからない・実際にファイルが置かれている端末はファイルの内容がわからない

○　Winny 利用の実態

・著作権あるデジタルコンテンツ（音楽・映像）の交換

・サーバが無いので規制が難しい

→2004 年 5 月に開発者が著作権侵害幇助の疑いで逮捕・起訴

○　起訴した側の主張

・現行の法律から鑑みて、開発者の行動は著作権侵害を幇助するという意味で罪である

○　開発者の支援者や技術者の主張

・技術の進歩とともに法律も進化しなくてはならず、現在の技術によって簡単に著作権法違反が発生してしまう現状の方が問題である

○論争における論点の幅が大きい

・現実の著作権侵害をどう解決すれば良いのか？

・侵害目的でモノを作るのは犯罪に問えるか？

・モノを作る人間は，使われ方にまで責任を負うか？

・匿名システムはプライバシー保護につながるが、それを一切禁じることにならないか？

・良い流通システムが無いからファイル交換ソフトが使われたのではないか？

＜１０－３－２　プライバシーとセキュリティの境界＞

（b）個人情報保護法

○　個人情報保護法（2003 年月成立、20005 年４月施行）

・個人情報の有用性に配慮しながら個人の権利利益を保護することが目的

（以下は参考までに）

＜個人情報取り扱い事業者が守るべきルール＞

1. 利用・取得に関するルール,

・ 個人情報の利用目的をできる限り特定し，利用目的の達成に必要な範囲を超えて個人情報を取り扱うことを禁止

・ 偽りその他不正な手段によって個人情報を取得することを禁止

・ 本人から直接書面で個人情報を取得する場合には．あらかじめ本人に利用目的を命じする必要があり．間接的に取得した場合には．すみやかに利用目的を通知または公表する必要がある

2. 適正．安全な管理に関するルール

・ 顧客情報の漏洩などを防止するため，個人データを安全に管理し，従業者や委託先を監督する必要がある

・ 利用目的の達成に必要な範囲で，個人データを正確かつ最新の内容に保つ必要がある.

3. 第三者提供に関するルール

・ 個人データをあらかじめ本人の同意を取らないで第三者に提供することは原則禁止.

4. 開示等に応じるルール

・ 事業者が保有する個人データに関して，本人から求めがあった場合は．その開示・訂正・利用停止などを行なわなければならない

・ 個人情報の取り扱いに関して苦情がよせられたときは．適切．迅速に処理しなくてはならない.

（c）セキュリティ

○　情報セキュリティ：情報システムにおける安全性確保のこと。

以下の３つの側面を持つ。

３つの側面	説明	侵害された場合
機密性（Confidentiality）	認可されたもののみ情報アクセス可能にすること	情報漏洩
完全性（Integrity）	情報が正確でかつ完全なこと	情報改ざん
使用可能性（Availability）	必要な時に必要な情報資源にアクセスできること	使用妨害

○　**不正アクセス禁止法**

・不正アクセスの禁止

例）他人のアカウントとパスワードの無断使用、セキュリティホール使用による侵入

・不正アクセスを助長する行為の禁止

例）他人のアカウントとパスワードを第三者に提供すること

・罰則、再発防止のための措置などを定める

・アクセス管理者、不正アクセス防御措置を講じる努力義務

・防御措置に対する都道府県公安委員会による援助について定める

→　高度通信社会の健全な発展に寄与

（d）セキュリティ確保の技術的枠組み

○　個人認証技術

・情報システム管理者がアクセスしてきた人間を、利用権者であるかどうかを識別する技術

例）**アカウント ID**、パスワード、生体固有な特徴による個人認証

○　暗号化技術

・元のデータを解読できないデータに変換すること（第４章）

○　公開鍵暗号方式と **PKI(Public Key Infrastructure)**　（第４章）

○　電子署名（第４章）

○公開鍵暗号方式

送信者：受信者が公開している公開鍵を入手し、その鍵で暗号化を行なった結果を送信

受信者：送信者から送られてきたデータを自分の秘密鍵で複合し、元のデータに戻す

○PKI

・公開鍵暗号方式を利用したセキュリティインフラストラクチャーのこと

・各人に固有の正しい公開鍵を配布＋各人の身分を証明する証明書を発行

＝認証局が行う

（e）セキュリティとプライバシー

○国境を越えての情報流通

→国ごとの「セキュリティ文化」が争いの元に

・セキュリティ文化が異なると、プライバシー、所有権、犯罪、賠償責任など多くの事柄で調整が必要

・セキュリティ文化：情報システム及びネットワークを開発する際にセキュリティに注目し、また、情報システム及びネットワークを利用し、情報をやり取りするにあたり、新しい思考や行動の様式を取り入れること

§４　インターネットと民主主義

・サイバースペース：ネット上の自由で規制のない議論空間

・情報技術が民主主義を促進する可能性が示唆された

＜10－４－１　インターネットは民主主義を加速させるか＞

○インターネットのあり方と民主主義

（１）IT は多対多の通信を可能にする

（２）「情報は力である」という考え

（３）かつて存在していた団結に対する障壁を打ち壊し、弱者に力を付与

→民主的な手続きや制度を支援する形で、情報技術は構築される必要がある

○　アラブにおける民主化

→・デモの動員にインターネットが貢献

・インターネットが普及している国ほど、デモ前に Tor（アクセス匿名化ツール）が多く利用される傾向

→インターネットが民主化を促進する要素を持つことは否めない。しかし、実際には民主化とは正反対のことも。

（例：電子会議室における荒らし、無責任な書き込み）

・情報技術は民主的な手続きを拡張する可能性を持つ

・しかし、その可能性を現実にするには規則の整備が必要

例）運営のルール、参加の資格と責任、結果を行政に反省させるルールなど

→民主的な手続きや制度を支援する形で、情報技術は構築される必要がある

＜10－4－2　ネットは公共空間か共同体か＞

公共空間	共同体
誰もがアクセスしうる	閉じた領域
成員のもつ価値は互いに異質	成員が同じ価値を共有
人々の間に生起する共通の関心が重要	内面に抱く情念(愛国心など)が統合の媒体
一元的・排他的な帰属を求めない	アイデンティティ(自己同一性)をもつ

→インターネットは、血縁・地縁などの閉鎖空間から離れたオープンスペースを作る可能性を持つ

○　公共空間と共同体

・国際比較研究によると、日本は他国に比べ「ブログは社会運動の場」と考えるブログ管理者が少ない

→日本のブログ管理者は公共空間ではなく「共同体」を作っている可能性

・オープンスペースを作っても、ネット上での新たな共同体間のギャップと分断が作られるのでは、公共性からは程遠い。

＜10－4－3　ネットの功罪＞

○　大津市中2いじめ自殺事件（2011 年）

・大津市内の中学 2 年生がいじめを理由に自殺

・教育委員会は生徒へのアンケート調査結果を公表せず、被害届も受理拒否

・アンケート結果に関する新聞報道を受け、ネット上で「炎上」

→ネットとマスコミの相互作用で全国的議論となり、地域の権力構造を覆した

○　情報技術の３つの特徴が広範囲の倫理的諸問題を引き起こす

・グローバルで多対多の通信射程

→短い時間とわずかな努力で他人に害を及ぼしうる

・匿名性

→不可視性の感覚が、多くの人々に他の仕方では行われないかもしれない行動を自由に行わせる

・複製可能性

→上記の問題を悪化させる

→**人権侵害、肖像権侵害、著作権侵害、プライバシー侵害**を誘発、深刻化