# 第3章
# コンピュータとディジタル情報

## 第2節　情報のディジタル化

## 情報のディジタル化の方法

コンピュータはデータをどんなルールで2進数にするのか？　その方法を個別に学習する。

基本は
**1対1対応**

| | |
|---|---|
| 数値のディジタル化 | 正の整数，負の整数，浮動小数点表示 |
| 文字のディジタル化 | 文字コード，1バイト文字，2バイト文字 |
| 音声のディジタル化 | 標本化，量子化，符号化，周波数，D-A変換，A-D変換 |
| 画像のディジタル化 | 標本化，量子化，符号化，解像度，3原色<br>静止画，動画，ペイント系，ドロー系 |
| データの容量と圧縮 | 圧縮，可逆性，非可逆性，解凍 |
| ファイル形式と拡張子 | 品質，特徴 |

## 数値のディジタル化は3パターン

関連：p44-47

問い：4bitで，何個の数を表現できるか。　参考 https://ja.wikipedia.org/wiki/数の比較

問い：64bitで，何個の数を表現できるか。

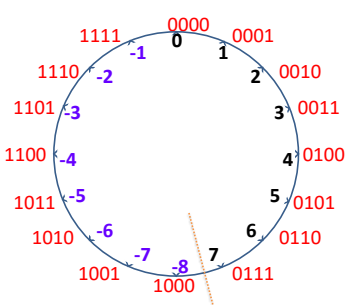
実習（別紙）：3とおりの方法で，4ビットの数を表現してみよう。
- 固定小数点表示（符号なし整数，符号付き整数）
- 浮動小数点表示（仮数部×基数$^{指数部}$ の形式）

固定小数点表示（整数型）
- 符号なし整数　（例　0，1，2，3，・・・）
- 符号付き整数　（例　・・・，−2，−1，0，1，・・・）

浮動小数点表示（実数型）
- 仮数部×基数$^{指数部}$ の形式　（例　$6.02\times10^{23}$）

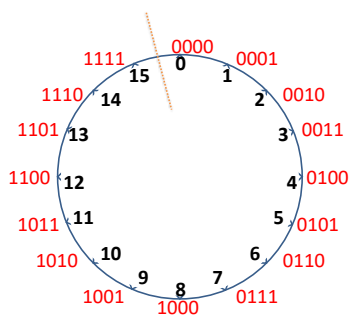
基本は，
**0と1の並び**
（ビット列）と
**表現するもの**
との
**1対1対応**

## 符号なし整数と符号付き整数を図示




**10000**

## 符号なし整数と符号付き整数を図示




## 数値のディジタル化の実際

基本は
**1対1対応**

よく使われる32bitだと次の通り
- 2進数32桁 ⇔ $2^{32}$（= 42億9496万7296 ）通りの状態を表せる。

固定小数点表示（整数型）
- 符号なし整数（0〜42億9496万7295まで）
- 符号付き整数（-21億4748万3648〜+21億4748万3647まで）

浮動小数点表示（実数型）
- 符号部 & 仮数部 × 基数$^{指数部}$ の形式（例：$-1.23456\times10^{-12}$）

□　□□□□□□□□　□□□・・・23個並ぶ・・・□□□□□
符号部　指数部(8bit)　仮数部(23bit)

メリット：有効数字を適切に保ちながら幅広く実数を表現できる

## 小数や大きい数（概数）はどのように表現する？

| | | | | | |
|---|---|---|---|---|---|
| 0000 | 0001 | 0010 | 0011 | 0100 | ・・・・ |
| ・・・・ | 1011 | 1100 | 1101 | 1110 | 1111 |

これらを　± 1.□ × 2^□　の形だと考える。
符号　仮数　×基数(底)　指数

（つまり □□□□を 符号・仮数・指数 に振り分ける）

- ±について・・・・0⇔＋　　1⇔－
- 「1.」と「×2」　　・・・・表現する必要なし
- 0は表現できない等のため特別ルール必要

※ 実習プリントで，ざっくりと感覚をつかんでみよう。

## 文字のディジタル化

関連：p48

基本は
**1対1対応**

コンピュータにおける文字の処理
文字を入力するとコンピュータは，
⇒ 規則に従って，文字と**ビット列**（0,1の並び）を**1対1対応**させる。
⇒ データ保存は，ビット列の状態で。
⇒ 表示・印刷には，ビット列に対応した文字の形を表示させる。

文字集合：　文字の一覧
文字コード：　文字に対応したビット列
文字コード体系：　上記対応の一覧
文字エンコーディング（符号化処理）：
　文字をビット列に置き換える処理
　（※　元に戻すことが**デコーディング**）

| 文字 | 16進数 | ２進数 |
|---|---|---|
| H | ４８ | 0100 1000 |
| e | ６５ | 0110 0101 |
| l | ６Ｃ | 0110 1100 |
| l | ６Ｃ | 0110 1100 |
| o | ６Ｆ | 0110 1111 |

## 文字コード体系の例

関連：p48

ASCII と JIS


問い：コード表を使って Kobe を，16進数と2進数で表そう（ノートに）。答はあとで・・・。

## 文字のディジタル化の具体例

関連：p48-49

1バイト文字（半角文字）　英字や数字やよく使う記号は1バイト（256種類）で十分。
カタカナも含めてもOK（このカタカナを**半角カタカナ**という）。

2バイト文字（全角文字）　日本語をすべて表示するには2バイト（65,536種類）必要。
（全角文字の英字・数字・記号・カタカナも作られた）

### 文字コード体系の種類

① ASCII　（情報交換用米国標準コード1963）　7ビット
② JIS　（ISO-2022-JP　メールで使われる）　8,16ビット
③Shift_JIS（Microsoft社が策定。Apple社でも使用）　8,16ビット
④ EUC-JP　（Extended Unix Code：UNIX,Linuxで使われる）　8,16ビット
⑤ Unicode（世界各国の言語に対応）　2バイト以上の場合もあり

21

## 実験：文字のディジタル化を確認する

関連：p48-49

問い：コード表を使って，Kobe という言葉を，
16進数と2進数で表してみよう。

| | | |
|---|---|---|
| K | 4B | 0100 1011 |
| o | 6F | 0110 1111 |
| b | 62 | 0110 0010 |
| e | 65 | 0110 0101 |

### 手順

1. マイドキュメント（ドキュメント）を開く。
2. 文字入力.txtをダブルクリック。
3. Kobe(改行)コ(改行)コ(改行) (改行)と入力して保存。
4. Bzエディタをダブルクリックして開き，ファイル「文字入力.txt」をドラッグ &ドロップ。
5. 文字コードの確認（教科書か画面かで）。
6. BZでリードオンリーを外して，数値を書き換える。
7. IMEパッドでシフトJISを見て，数値を確認。

22

## 文字化け・・・・3つの原因

関連：p48

- 半角カタカナの使用
- 機種依存文字の使用
- 文字コードの指定間違い

**機種依存文字対応表**
http://www.geocities.co.jp/SiliconValley-SanJose/5148/moji/

**キャリア6社の絵文字がついに統一**
http://k-tai.impress.co.jp/docs/news/20140424_646054.html

**絵文字**
http://www.softbank.jp/mobile/service/mail/3g/pictgram/

## 参考：IMEパッド 文字一覧（シフトJISの一部）

## 文字の表示・印刷

関連：p50

**グリフ** 文字の形（デザイン）　**フォント** グリフの一覧

- **等幅フォント** 文字幅が均一化（統一）されている
- **プロポーショナルフォント** 文字によって文字幅が異なる
- **ビットマップフォント** 点の配置関係でグリフを決定する
- **アウトラインフォント** 輪郭線を数式にしてその都度計算してグリフを決定する

**pt（ポイント）** 文字サイズの単位　1pt ＝ 約0.35mm

http://culture.cc.hirosaki-u.ac.jp/english/utsumi/info/moji_c6_ja.html

32

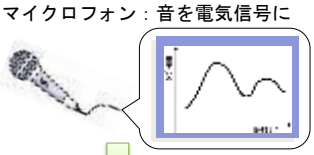
## 音声のディジタル化

関連：p51-52

音：空気の振動の波

空気の圧力が鼓膜をふるわせる。

マイクロフォン：音を電気信号に



電気信号を**ディジタル**にしたい！！

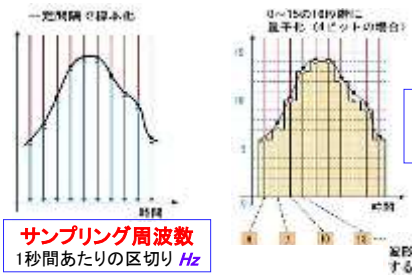
**A／D変換**　アナログ（電気信号）をディジタルに

**D／A変換**　ディジタルをアナログへ

## 波形のディジタル化（PCM方式）

関連：p51-52

**PCM(Pulse Code Modulation)方式**
**標本化**（サンプル取得）⇒ **量子化**（一定段階値に変換）⇒ **符号化**（2進数表現）

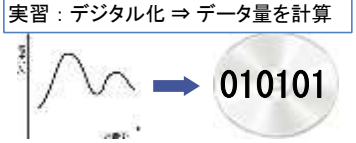


**サンプリング周波数**
1秒間あたりの区切り *Hz*

**量子化ビット数**
段階の桁数
（2進数で）*bit*

## 音声のA-D変換（実習とまとめ）

実習：デジタル化 ⇒ データ量を計算




1秒に何回標本化？
何段階？

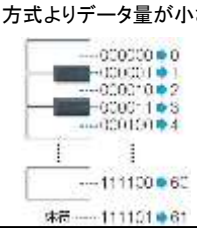
標本化定理　もとの音の最高周波数の2倍以上で標本化。
人の可聴帯域　約20Hz～20,000Hz。

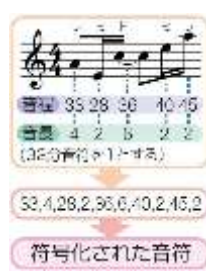
音楽CD　サンプリング周波数：44,100Hz　（1秒間に44100回）
量子化ビット数：16bit=$2^{16}$=65536段階　（0～65535）
チャンネル数：2ch　（マイク2つ）

## 楽譜のディジタル化（MIDI）

関連：p52

- PCM方式・・・・波形のディジタル化
- MIDI規格・・・・楽譜のディジタル化
  (Music Instrument Digital Interface)
  - 音源・音程等の変更が容易。
  - PCM方式よりデータ量が小さい。





## 色の作り方

関連：p56

●原色　種々の色を作るもとになる色

●光の３原色（ＲＧＢ）

赤（Red）　緑（Green）　青（Blue）
光を重ねる ⇒ 白へ（加法混色）


●色の３原色（ＣＭＹ）

シアン（Cyan）　マゼンタ（Magenta）　イエロー（Yellow）
インクを重ねる ⇒ 黒へ（減法混色）


• ３色の強さを変えて，多くの色を出す

## 画像のディジタル化の基本

関連：p53

●ペン先は細いほど，微細な表現が可能

●色数は多いほど，現実に近づけることが可能

大事なキーワード
- ◆ 解像度・・・・・どれほど細かく描けるか？
- ◆ 階調・・・・・・・色を何段階で表せるか？

思い出せ！　それが次の発想につながる

◆音のディジタル化：　**波の高さ**を何段階かにして，2進数にして，**サンプルの個数**だけ並べた。 ⇒　並んだ**0**と**1**の個数がデータ量(情報量)だった。

◆画像のディジタル化：　**色を何段階か**にして，2進数にして，色のついた**点の個数**だけ並べる！・・・という発想。

## 階調（モノクロの場合）　何色使う？

関連：p55

●階調　濃淡の段階


ひとつの画素に対して，0か1を8個ずつ使う。

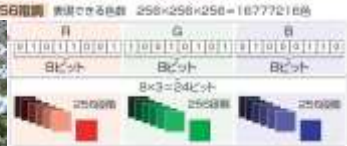

ひとつの画素に対して，0か1を1個だけ使う。

よく使われるのはモノクロではなくて，
カラーでしょ？？？

## 階調（カラーの場合）

関連：p54


3bitで8色




24bitで1678万色

## 階調（まとめると）

関連：p55

● 階調　濃淡の段階

**世の中でよく使われるのは・・・**

● 赤 緑 青：各8bit（$2^8$通り）
　すなわち 256階調（0～255）

● 256×256×256 ＝ 約1678万色

これを
フルカラー（24ビットカラー）
という。

## 解像度・画素（ピクセル）　絵の細かさ

関連：p54

| | |
|---|---|
| 画素（ピクセル） | （ディスプレイ等で）画像を構成する点の最小単位。 |
| 解像度 | 1インチ（2.54cm）の分割数（画素の密度）。ｄｐｉで表す（dots per inch） |
| 画面解像度は，ｐｐｉ（pixels per inch）で表すこともある | |

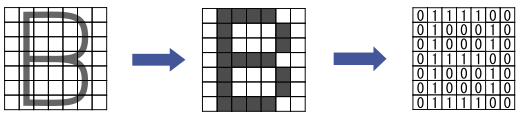
## 画像のディジタル化の手順

関連：p53

**標本化** 画素に分割し，濃淡（標本値）を取り出す。
**量子化** 標本値を，段階（階調）の整数値に近似する。
**符号化** 0と1に置き換える。

**例**
**（白と黒で）**

- 標本化　格子状に分割。
- 量子化　黒か白か（２段階），判定する。
- 符号化　例えば，**黒**を**1**に，**白**を**0**にする。



0111100 0100010 0100010 0111100 0100010 0100010 0111100

## ビデオで復習・・・

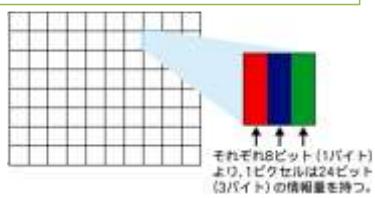
**・NHKの番組（2010年） 7分30秒**
**・16進数を使わず10進数で説明（画像⇒音⇒文字）**

マス目 1000万以上(http://kakaku.com/camera/digital-camera/itemlist.aspx?pdf_Spec202=1600-)
色の種類 約1700万(24bitカラー)

## 画面のデータ量の計算

関連：p53～56

教室のディスプレイは 1920×1080ピクセルで，フルカラー表示である。データ量(情報量)は何バイトか？　それは約何メガバイトか？

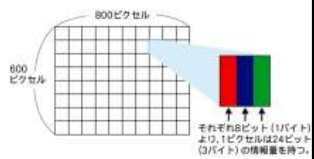



赤,緑,青が各1バイト（8ビット，256段階）の情報を持つ。
1920 × 1080 × 3（バイト）＝ 6,220,800（バイト）（49,766,400ビット）
総ピクセル数
約 5.93 MB

## 画像と文字のデータ量比較

関連：p53～56

例：800×600ピクセルのフルカラー画像のデータ量について考えてみよう。




*（計算を簡単にするために，単位は1024ではなく1000を使って）*

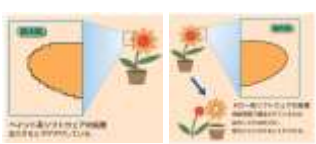
① 容量1.44MBのフロッピーディスク（FD）に，800×600ピクセルの（上例の）画像は，約何枚くらい保存できると考えられるか。
② FDに，1ページあたり40字×18行＝720字の日本語（2バイト文字）が印刷された文庫本の文字情報は，約何ページ保存できると考えられるか。
③ コンピュータにとって，どちらの処理がしんどそう？？

## 画像処理（ペイント系・ドロー系）

関連：p57

図形・画像の扱い方：**2種類**

フォントの種類，覚えている？

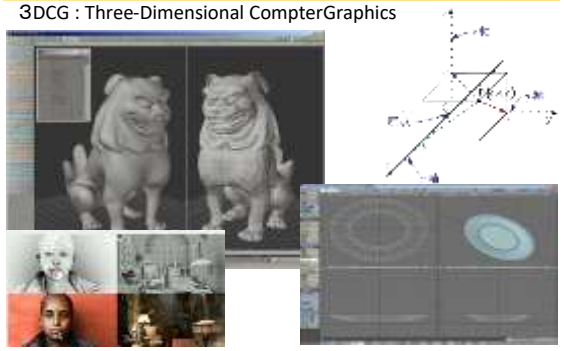


| | ペイント系（ピクセルグラフィクス） | ドロー系（ベクターグラフィクス） |
|---|---|---|
| 処理方法 | 図形を点の集合で表現（**ラスター表現**） | 図形を形状データ(座標・直線・曲線等)で表現（**ベクター表現**） |
| 拡大 | 拡大するとギザギザ | 拡大してもなめらか |
| 用途 | 写真等，グラデーションや微妙なニュアンスの表現に適する。 | イラスト・設計図等に適し，変形が容易。 |
| 特徴・他 | | 図形を部品に分割できる |

## 3次元コンピュータグラフィックス（3DCG）

関連：p57

3DCG : Three-Dimensional CompterGraphics



## 3次元コンピュータグラフィックス（3DCG）

関連：p57

3DCG : Three-Dimensional CompterGraphics

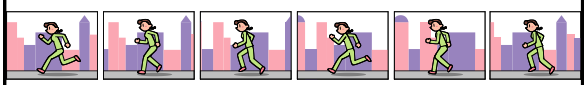
**モデリング（modeling）**
形状をつくること。
（面をポリゴンで表現）

**レンダリング（Rendering）**
画像を生成すること。
（質感や陰影等を計算）

## 動画のしくみ

関連：p58

少しずつ変化した静止画を次々と見せる。



- 動画の原理：残像効果を利用
  （パラパラまんがと同じ）
- テレビ：毎秒３０枚　(30fps)
  （又は60fps）
- 映画：　毎秒２４枚　(24fps)
- データ量を減らす工夫が必要

## データの容量と圧縮・解凍

関連：p60

・よりアナログに近い状態でディジタル化したい　⇒　データ量が非常に増加！

データの**圧縮**：ファイルのデータ量（ファイルサイズ）を小さくすること。

| 可逆圧縮 | 非可逆圧縮 |
|---|---|
| ●元データに戻せる | ●元データに（完全には）戻せない |
| ●例：ランレングス圧縮<br>run length encoding | ●例：音…　聞こえない周波数の音をカット<br>大きな音に隠れた小さな音をカット |
| | ●例：画像…色の違いを間引く |
| | ●例：動画…変化した部分（**差分**）だけを記録 |

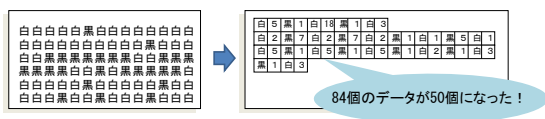
データの**解凍**（**展開・伸張**）：使用するためにもとの状態に戻すこと。

## 可逆圧縮のしくみ（例：ランレングス圧縮）

関連：p60


① 原稿を点として読み込む　② 点データを別表現にする

**ランレングス圧縮**：　同じデータが連続する場合に有効。
「データ」と「その長さ」を表現。（ファクシミリで使われる）




## ファイル形式と拡張子

関連：p61～63

**拡張子**　Windowsでは，ファイル名の，ドットより右の文字で
どんなファイルか（ファイル形式）を判断する。

| | | |
|---|---|---|
| 画像 | ●**BMP** | 無圧縮 |
| | ●**JPEG** | 圧縮率を変更できる　写真に適する（非可逆） |
| | ●**GIF** | 使用できる色数は最大256　イラストに適する（可逆） |
| | ●**PNG** | GIFの代替として規格された（可逆） |
| 動画 | ●**AVI** | 無圧縮や様々な圧縮があり |
| | ●**MPEG** | MPEG1，MPEG2，MPEG4等の規格がある（非可逆） |
| 音 | ●**WAVE** | 無圧縮　拡張子 wav |
| | ●**MP3** | WAVEの約10分の1のサイズ（非可逆） |

## ファイル形式の違いによるデータ量を調べる（実習）

実習：BMPファイル(写真とイラスト)を様々な形式で保存して，
結果（データ量や見た目の変化）を考察する。

① [レポート等提出]⇒[201611-1_画像圧縮x組]
⇒[201611-1_ファイル形式(xxx)]フォルダを開く。
② 名前を付けて保存を6回。　④ファイルを観察。

| | | BMP(元) | JPEG | GIF | PNG |
|---|---|---|---|---|---|
| 写真 | ファイルサイズ<br>特徴等 | | | | |
| イラスト | ファイルサイズ<br>特徴等 | | | | |