

# 111.IT基礎

## コンピュータの構成要素

- コンピュータの5大装置

- 入力装置・出力装置(I/O)

- 記憶装置(主記憶装置：メモリ)

- RAMとROM

- RAMの種類

- 制御装置・演算装置(プロセサ、CPU)

## ソフトウェアとプログラム

## OSの仕組み

- OSの目的

- OSの機能

## ハード・OS・アプリケーションの関係

## ビットとバイト

## 2進数

- 2進数の計算(基数変換)

- 2進数の計算 負の数場合

- 2進数の計算 小数点が入った場合

- 2進数で0.1を表現できるか？

## 16進数

## 文字コード

## 論理演算

- 真偽値(真理値)

- 集合

- 帰属と包含

- 命題と真偽値

- 基本的な論理演算

- 基本的な論理演算の組合せ

## 目的：

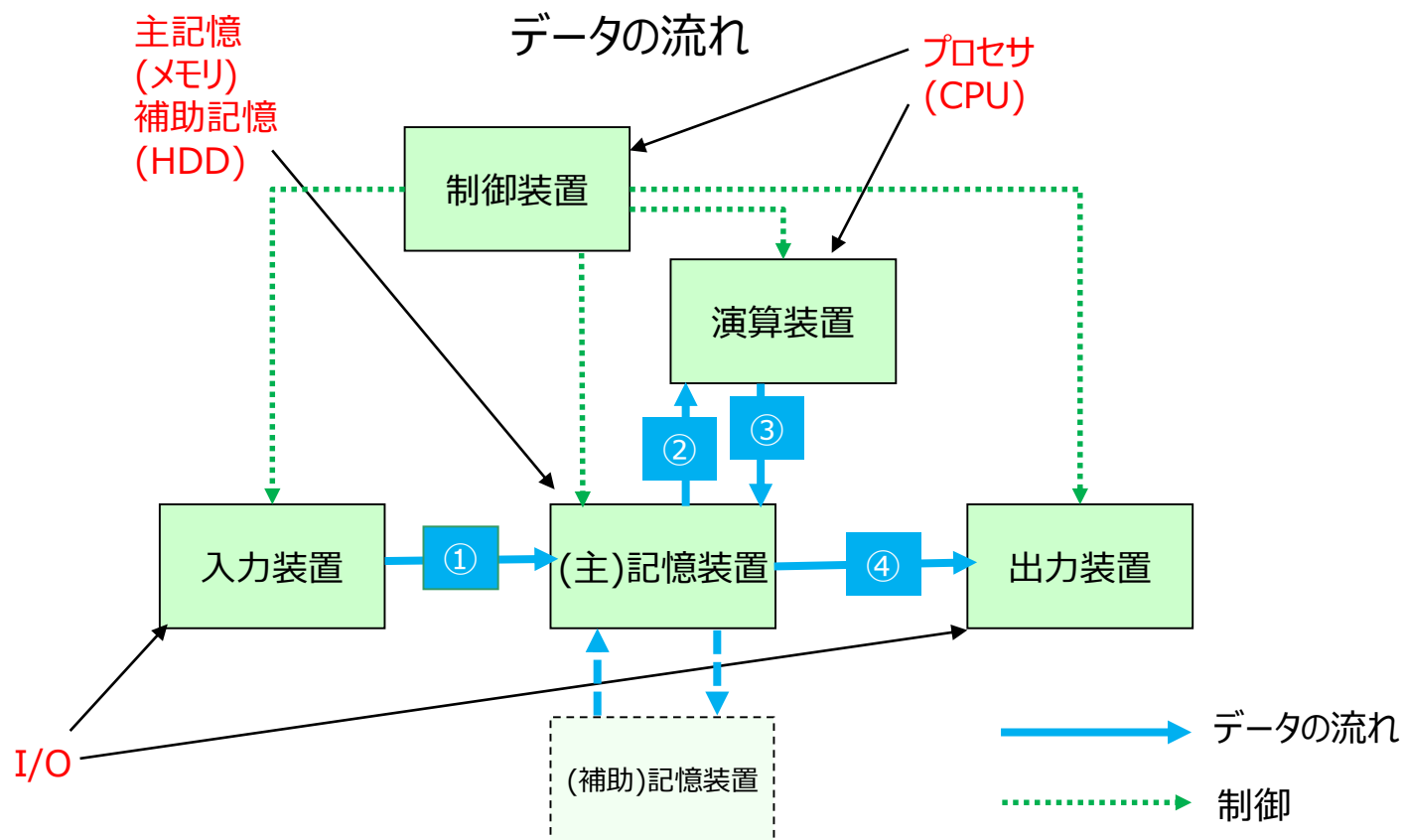
- 普段使用しているコンピュータについて、どのような要素で構成されているのかを学ぶ
  - ◆ ハードウェアとは？
  - ◆ ソフトウェアとは？
  - ◆ コンピュータではデータをどうあつかう？  
コンピュータの処理で欠かすことの出来ない2進数、16進数がどのような物なのか理解し、10進数との相互変換等を学ぶ
  - ◆ コンピュータの計算の一つ、論理演算がどのような物なのか学ぶ

## ゴール：

Javaを学習する上で情報処理に必要な基礎を理解する

## コンピュータの5大装置

- 入力装置
- 出力装置
- 記憶装置
- 演算装置
- 制御装置



※一般的にノイマン型(von Neumann architecture)と呼ばれる。

演算装置と制御装置をまとめた装置をCPU(中央処理装置)と呼ぶが、apple M2の様なSoC(System on a Chip)もある。

## 入力装置・出力装置(I/O)

### ■ 入力装置

外部からコンピュータに処理すべき情報（データ）やコマンドなどを**入力**する装置。

主な入力装置

キーボード、マウス、ペンタブレット、トラックボール、スキャナなど

### ■ 出力装置

コンピュータから外部に、処理した結果（データ）やコマンドの応答などを**出力**する装置。

主な出力装置

プリンター、モニタ、スピーカーなど

データやコマンドの出入り口である2つの装置を合わせて入出力装置という。入出力 (Input/Output)を略して**I/O装置**ともいう。

## 記憶装置(主記憶装置：メモリ)

主記憶装置は**メモリ(メインメモリ)**とも呼ばれ、処理手順を記載した**プログラム**や、処理前や処理後の**データ**などを、一時的または半永久的に記憶（保存）する装置。

現在は一般的に、半導体素子によって構成された半導体メモリを利用し、コンデンサの電荷(-)のあるなしで"0"/"1"を判断する。

## RAMとROM

メモリは体表的な種類としてRAMとROMがある。

### ■ RAM(Random Access Memory)

読み込みと書き込みができる。電源を切ると記憶が消える。

→揮発性

### ■ ROM(Read Only Memory)

(基本的には)読み込みのみできる。電源を切っても記憶が消えない。

→不揮発性

## RAMの種類

### ■ DRAM(Dynamic RAM)

コンデンサ(蓄電器)とトランジスタ(半導体)で構成され、コンデンサに電荷が溜まっているかないかで"0"か"1"かを判断する。自然放電してしまう為、一定間隔でコンデンサを再充電(リフレッシュ動作)する必要がある。構造が単純な為、集積度が高く、単価が安い。アクセス速度は遅い。

主に主記憶装置に用いられる。

### ■ SDRAM(Synchronous DRAM)

システムバスに同期して動作するDRAM。

PCの主記憶装置の標準規格である DDR(Double Data Rate SDRAM)、DDR2、DDR3、DDR4へ発展する

### ■ SRAM(Static RAM)

フリップフロップなどで構成される。DRAMの様なリフレッシュ動作は不要。構造が複雑なので、集積度は低く、単価は高い。アクセス速度は高い。

主にキャッシュメモリに用いられる。※またレジスタもフリップフロップなどで構成される記憶装置



## 制御装置・演算装置(プロセサ、CPU)

CPU(Central Processing Unit:中央処理装置)は制御装置と演算装置から構成される。

### ■ 制御装置

記憶装置・I/O装置・演算装置に対して制御を行う装置。

### ■ 演算装置

制御装置の指示に従って演算（計算）を実行する装置。

※SoC(System on a Chip)の様に、一つのチップにCPU、GPU、メモリなどを統合する物もある。  
→省スペース、省電力、データ転送の高速化が期待できるため、モバイルなどに使われる

ソフトウェアとはコンピュータプログラムのこと。

コンピュータを構成する電子回路や周辺機器などの物理的実体をハードウェアと呼ぶのに対して、形を持たない手順や命令などをソフトウェアと呼ぶ。

OSなどは基本ソフトウェア(システムソフトウェア)といい、ExcelやWordなどはアプリケーションソフトウェアという。

## OSの目的

- **ハードウェアの抽象化**

メーカーなどが異なった様々なハードウェアに対して、同一のアプリケーションの動作を保障する。

- **リソース管理**

複数のアプリケーションソフトウェアを同時に利用する際に、互いに独立して動作できるように資源を管理する。

- **利用効率の向上**

複数のタスクを同時に実行する際に、資源割り当ての順番や処理の割り当て時間を工夫することで、全体のスループットを向上させる。

## OSの機能

- **API(Application Programming Interface) の提供**

ソフトウェアの機能を利用するための関数（道具）。

- **プロセス管理**

プロセスの生成・実行・制御・消滅を管理する。プロセス間通信や排他制御もプロセス管理の役割である。また、プロセスへのリソースの割り当てを行う。

プロセス・・・コンピュータにおける処理の単位。

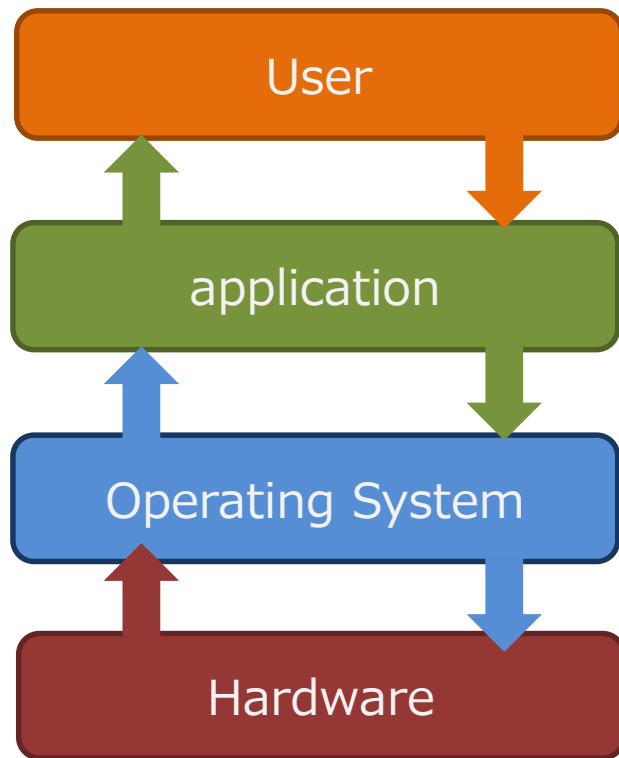
- **メモリ管理**

- **ファイルシステム**

ファイルシステムとは、記憶装置に記録されているデータを管理する方式。代表的なもので windows の NTFSやFATがある。

# ハード・OS・アプリケーションの関係

ハードウェア上にOSがあり、  
OS上に複数のアプリケーションがある。  
アプリケーションはハードウェアを意識しない。  
様々なハードウェアの違いはOSで吸収される。



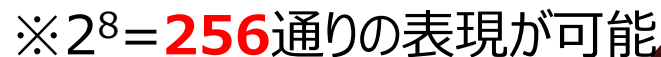
コンピュータは電子回路の組み合わせで作られており、電気で動いている。電子回路は電流のON(流れている)、OFF(流れていない)の状態、数や文字など様々なデータを扱う。

- **ビット(bit)**

CPUアーキテクチャやデータバス、ネットワークの通信速度などの単位はビット。



メモリやファイルサイズを表すときの単位はバイト。



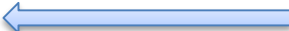





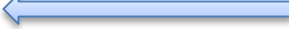
# 2進数



コンピュータは全ての処理を0と1の組み合わせである2進数で行う。人間が曰ごろ扱っている10進数は処理する事ができない。コンピュータの内部で10進数の値を2進数に変換して処理を行っている。

2進数は0と1だけで表す。

例).2進数での表し方

10進数の0	.....	0			1ビットで表せる
10進数の1	.....	1		桁が上がる	
10進数の2	.....	10			2ビットで表せる
10進数の3	.....	11		桁が上がる	
10進数の4	.....	100			3ビットで表せる
10進数の5	.....	101			
10進数の6	.....	110			
10進数の7	.....	111		桁が上がる	
10進数の8	.....	1000			4ビットで表せる

※一桁の2進数の数を1ビットと表現する。

## 2進数の計算(基数変換)

### 基数

2進数の2や、10進数の10を基数と呼ぶ。

数の各桁の重み付け基本となる数。

### 基数変換

○進数で表記された数を、△進数の表記に変換する。(基数を変換する)

→値自体は変わらない

### 桁の重み

2進数や、10進数など、各記数法での数の各桁の単位。

n進数の場合、各桁の重みは次のようになる

$n^3$	$n^2$	$n^1$	$n^0$	.	$n^{-1}$	$n^{-2}$
1000の位	100の位	10の位	1の位		-1の位	-10の位
2	9	8	7	.	1	2

### べき乗(底と指数)

$10^2$ (10の2乗) ……10の部分を底(基数)、2の部分を指数と呼ばれる二つの数で定まる算法

※累乗…… 指数が自然数・整数の場合

※べき乗…… 指数が自然数・整数以外でも良い



## 2進数の計算(基数変換)

10進数の数、2進数の数を、桁の重みで分解する

※10進数の123を桁の重みで分解

$$\begin{array}{rccccccc} 123 & = & 1*100 & + & 2*10 & + & 3*1 \\ & & \downarrow & & \downarrow & & \downarrow \\ & & 1*10\text{の}2\text{乗} & + & 2*10\text{の}1\text{乗} & + & 3*10\text{の}0\text{乗} \end{array}$$

**桁の重み(各桁の重みの基本)**

n進数のx桁の数値は、nを底(基数)とした べき乗で表す事ができる。

→底(基数)がn、指数が(x-1)の べき乗で表せる

※2進数の101を桁の重みで分解

$$\begin{array}{rccccccc} 101 & = & 1*100 & + & 0*10 & + & 1*1 \\ & & \downarrow & & \downarrow & & \downarrow \\ & & 1*2\text{の}2\text{乗} & + & 0*2\text{の}1\text{乗} & + & 1*2\text{の}0\text{乗} \end{array}$$

**桁の重み(各桁の重みの基本)**

## 2進数の計算(基数変換)

問題：

これらの底(基数)と指数の関係から、10進数から2進数への変換を考える。

①2進数の1111を10進数で表すと？

②10進数の13を2進数で表すと？

## 2進数の計算(基数変換)

問題：

これらの底(基数)と指数の関係から、10進数から2進数への変換を考える。

①2進数の1111を10進数で表すと？

基数で分解	$1111 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
各桁を計算	$= 8 + 4 + 2 + 1$
	$= 15$

②10進数の13を2進数で表すと？

「基数で割る」ことで、割った余りが1桁目からの値になる

$13 / 2 = 6 \cdots 1$	(6 余り1)余りが1桁目の値
$6 / 2 = 3 \cdots 0$	(3 余り0)余りが2桁目の値
$3 / 2 = 1 \cdots 1$	(1 余り1)余りが3桁目の値
$1 / 2 = 0 \cdots 1$	(0 余り1)余りが4桁目の値

13を2進数で表すと、1101

## 2進数の計算 負の数場合(※コンピュータ上での扱い)

一般的には「2の補数表現」で考える。

### 「-10」を「8ビット」の2進数で表現する場合

- ① 「10」を「8ビット」の2進数で表現する → 「00001010」
- ② 「00001010」を反転させる → 「11110101」
- ③ 「11110101」に「1」を加える → 「11110110」
- ④ 「11110110」が答えとなる(※00001010の2の補数になっている)

一般的に2進数で表現するとき、先頭が「1」であれば「負」であると考え(例).

- ・ 「11111111」 → 「-1」
- ・ 「10101010」 → 「-86」

先頭を確認すれば正負が分かるので、1番上のケタを「符号ビット」と言う

### 「11110110」を10進数で表現する場合

- ① 「11110110」の先頭から「1」を引く → 「11110101」
- ② 「11110101」を反転させる → 「00001010」
- ③ 「00001010」を10進数で表現する → 「10」
- ④ 「11110110」は先頭が「1」なので「10」に負の符号をつける → 「-10」

## 補数とは

計算したときに、桁上がりが発生する最小数(桁上がりが発生しない最大数)のこと。

桁上がりが発生する最小数を**基数の補数**、  
桁上がりが発生しない最大数を**減基数の補数**という。

例).

### ◆ 2進数

- 1の補数(減基数の補数)
- 2の補数(基数の補数)

### ◆ 8進数

- 7の補数(減基数の補数)
- 8の補数(基数の補数)

### ◆ 10進数

- 9の補数(減基数の補数)
- 10の補数(基数の補数)

例). 2進数

#### ■ 1の補数

- 「0001」の1の補数は「1110」 ( $2^4 - 1 = 15$ )
- 「0101」の1の補数は「1010」

#### ■ 2の補数

- 「0001」の2の補数は「1111」 ( $2^4 = 16$ )
- 「0101」の2の補数は「1011」

例). 10進数

#### ■ 9の補数

- 「7」の9の補数は「2」 ( $10^1 - 1 = 9$ )
- 「369」の9の補数は「630」 ( $10^3 - 1 = 999$ )

#### ■ 10の補数

- 「7」の10の補数は「3」 ( $10^1 = 10$ )
- 「369」の10の補数は「631」 ( $10^3 = 1000$ )

減基数の補数は、2進数のビット反転に、  
基数の補数は、減算を加算で計算する時に用いられる。

## 2進数の計算 小数点が入った場合

10.011を10進数の場合と2進数の場合で考える。

### 10進数の場合

10進数の重み付け(基数)は10

10.011

$$= 1*10 + 0*1 + 0*0.1 + 1*0.01 + 1*0.001$$

$$= 1*10^1 + 0*10^0 + 0*10^{-1} + 1*10^{-2} + 1*10^{-3}$$

### 2進数の場合

2進数の重み付け(基数)は2

10.011

$$= 1*10 + 0*1 + 0*0.1 + 1*0.01 + 1*0.001$$

$$= 1*2^1 + 0*2^0 + 0*2^{-1} + 1*2^{-2} + 1*2^{-3}$$

$$= 2 + 0 + 0 + 0.25 + 0.125$$

$$= 2.375$$

# 2進数



問題：  
10進数の0.1を2進数で表すと？

問題：

10進数の0.1を2進数で表すと？

$$2^{-1} \cdots 0.5$$

$$2^{-2} \cdots 0.25$$

$$2^{-3} \cdots 0.125$$

$$2^{-4} \cdots 0.0625$$

$$2^{-5} \cdots 0.03125$$

$$2^{-6} \cdots 0.015625$$

$$2^{-7} \cdots 0.0078125$$

$$2^{-8} \cdots 0.00390625$$

$$0.1 / 2^{-4} = 1 \text{ 余り } 0.0375$$

$$0.0375 / 2^{-5} = 1 \text{ 余り } 0.00625$$

$$0.00625 / 2^{-8} = 1 \text{ 余り } 0.00234375$$

.....

$$0.000110011001100\cdots$$

表せない



## 2進数で0.1を表現できるか？

- できない。いつまでも続く無限小数になる
- 2進数の無限小数は途中で切り捨てられる
- つまり **2進数による浮動小数点の計算は正確でない場合があることに注意！**

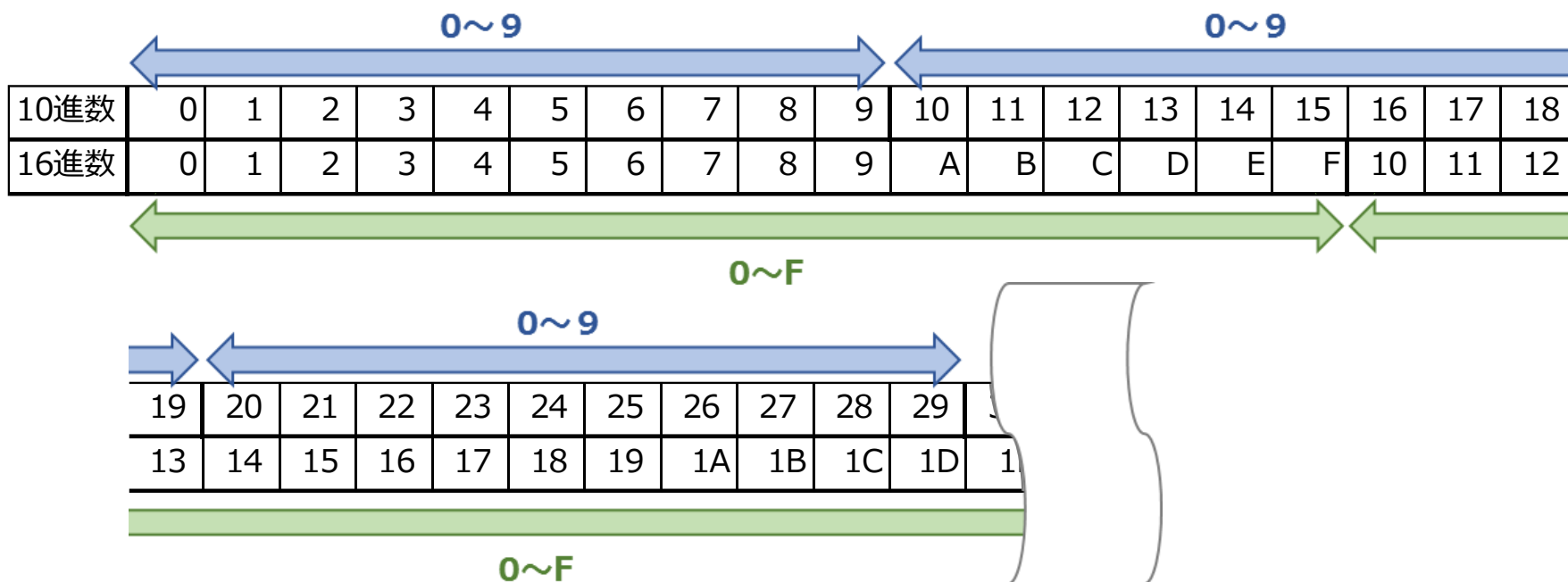
その他に、打ち切り誤差、情報落ち、丸め誤差など2進数を用いた計算では正確な計算結果を取得できない場合がある。

**Javaの計算は2進数で行われます。  
Javaでプログラミングをする以上、2進数の計算で起こる問題に意識する必要があります。**

# 16進数

2進数4桁（0～15）を1文字（0～9、A～F）で表現する。

「進数」の名前のとおり16で桁上がりする。



$$1F \cdots 1 \times 16^1 + 15 \times 16^0 = 16 + 15 = 31$$

$$CA \cdots 12 \times 16^1 + 10 \times 16^0 = 192 + 10 = 202$$

# 16進数



16進数2桁で1バイト(byte)を表すことができる。

プログラム上でデータなどを扱う時は、バイト(byte)つまり16進数で扱う。

1バイトで表せる範囲

00～FF

人間の世界では文字を利用している。  
日本であれば日本語の文字を利用している。  
この文字をコンピュータが利用・表現出来るように割り当てたバイト表現。

例)

Unicode表

[https://ja.wikipedia.org/wiki/Unicode%E4%B8%80%E8%A6%A7\\_0000-0FFFa](https://ja.wikipedia.org/wiki/Unicode%E4%B8%80%E8%A6%A7_0000-0FFFa)

開発の中で一般的に利用される文字コードには以下がある。

- ①JIS
- ②EUC-JP
- ③シフトJIS
- ④UTF-8

文字コードの話は難しいので、現状は1種類の文字を表す方法にも上記のようなものがある程度の理解で良い。

今回の研修で作成するプログラムは特に指示の無い限り「UTF-8」で作成・保存するように。

## 真偽値(真理値)

「真(true)」か「偽(false)」のどちらかの値。2つあわせてブール領域という。

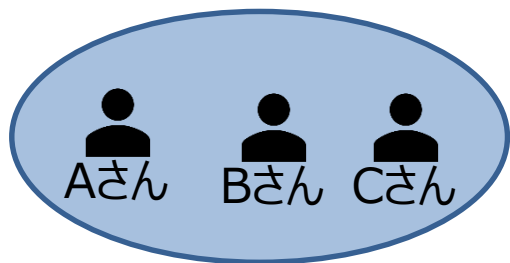
コンピュータでは1(true)、0(false)で表す。

→ 2進数で扱える

## 集合

物の集まり。集合を構成するものを元(要素)という。

(ブール領域も「真」「偽」2つの元の集合)



会社員

「会社員」という集合

「Aさん」「Bさん」は「会社員」という集合の元(要素)

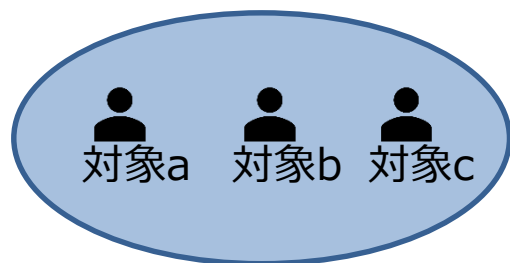
会社員を条件としたとき、

集合は条件を満たしている(真である)

会社員 = {A, B, C}

と書く

## 帰属と包含



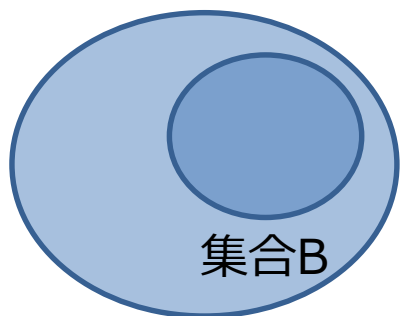
集合A

### 帰属

対象aが集合Aの元(要素)のとき、aは集合Aに属している。

$a \in A$

と表す。



集合A

### 包含

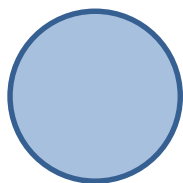
集合Aと集合BについてBに属する元がすべてAにも属する時、AはBを包含している。またBはAの部分集合である。

$B \subset A$  または  $B \subseteq A$

と表す。

## 命題と真偽値

集合Aは  
命題①を満たす  
(真である)



集合A

命題①を満たさない  
(偽である)集合を  
Aの補集合という



集合A

### 命題

判断(正しいか誤りか)が明確な文章や条件式を**命題**という。命題は**真偽値**をもつ。

青丸の内側を、命題①が真の集合Aとした場合、  
青丸の外側は、命題①が偽の集合でAの補集合という。

### 論理演算

2つ以上の真偽値から求める演算。

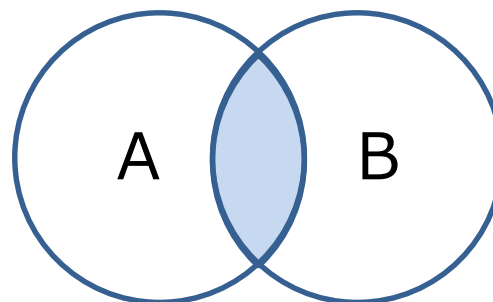


## 基本的な論理演算

### 論理積

A AND B ( $A \cdot B$ )

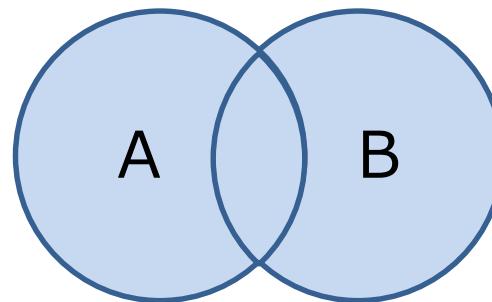
※ 積集合  $A \cap B$



### 論理和

A OR B ( $A + B$ )

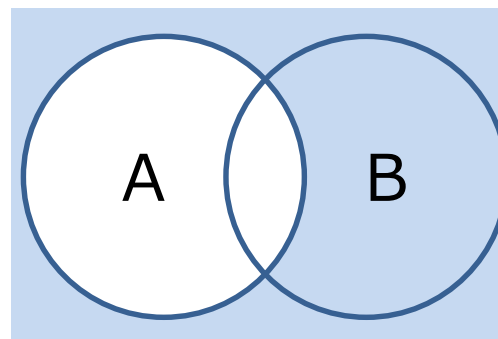
※ 和集合  $A \cup B$



### 否定

NOT A ( $\neg A$ )

※ 差集合  $U - A$   
(Uは全体)



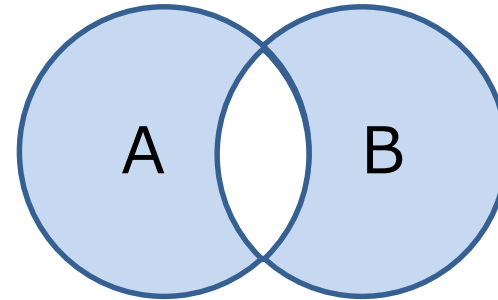
## 基本的な論理演算の組合せ

### 排他的論理和

A XOR B

$(\neg A \cdot B + A \cdot \neg B)$

※ABが異なる値の時true

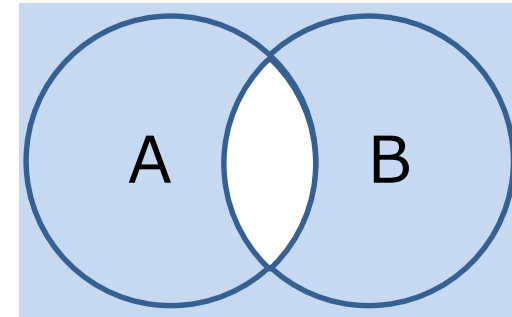


### 否定論理積

A NAND B

$(\neg(A \cdot B))$

※論理積の否定

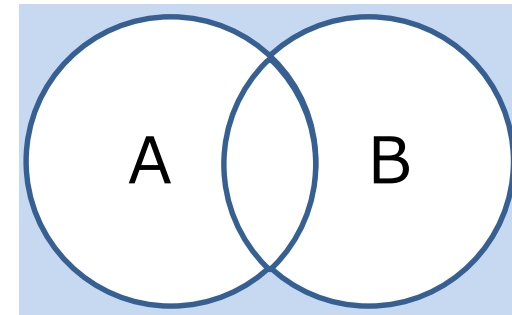


### 否定論理和

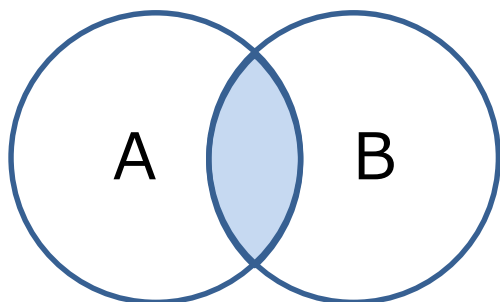
A NOR B

$(\neg(A + B))$

※論理和の否定



## 基本的な論理演算

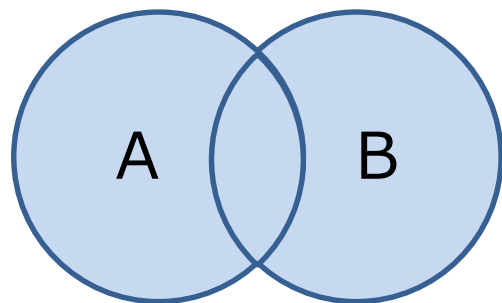


### 論理積

$A \text{ AND } B \ (A \cdot B)$

※ 積集合  $A \cap B$

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

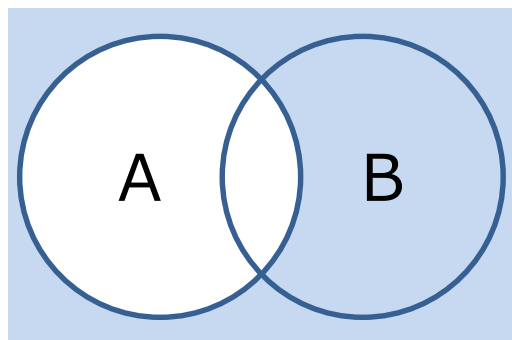


### 論理和

$A \text{ OR } B \ (A + B)$

※ 和集合  $A \cup B$

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1



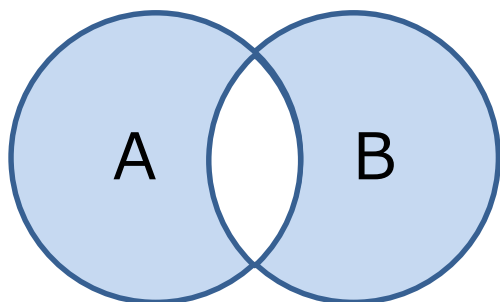
### 否定

$\text{NOT } A \ (\neg A)$

※ 差集合  $U - A$   
(Uは全体)

A	B	NOT A	NOT B
0	0	1	1
0	1	1	0
1	0	0	1
1	1	0	0

## 基本的な論理演算の組合せ



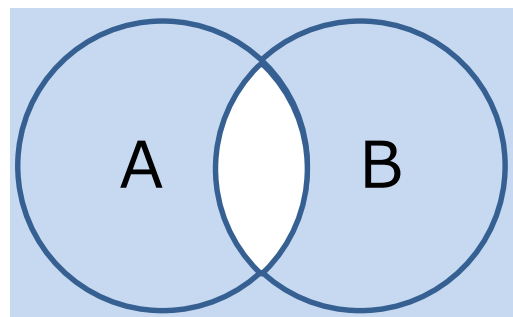
### 排他的論理和

$A \text{ XOR } B$

$(\neg A \cdot B + A \cdot \neg B)$

※ABが異なる値の時true

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



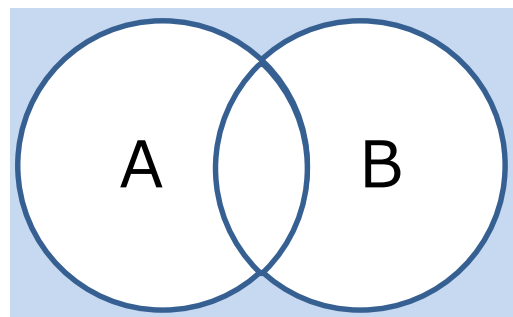
### 否定的論理積

$A \text{ NAND } B$

$(\neg(A \cdot B))$

※論理積の否定

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0



### 否定的論理和

$A \text{ NOR } B$

$(\neg(A + B))$

※論理和の否定

A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0