

プログラミング演習 A 教材 (#9) = これまでに出てきた機能などのまとめ =

□ データの型

char: 文字(character)を扱うための型。'x', 'a' など。

int: 整数(integer)を扱うための型。123, 0, -3 など。

float, double: 実数を扱うための型。123.456, 0.0001, 0.0, 1024.0 など。

123.456 は実数でも double 型。float 型と明記したい場合は 123.456f と最後に fをつける。

□ 代表的な printf の変換文字列

	意味	使用例	その結果
%c	文字を表示	<code>printf("[%c]\n", 'a');</code>	[a] 文字がそのまま表示される。
%d	整数を表示	<code>printf("[%d]\n", 10);</code>	[10] 桁数不定
		<code>printf("[%5d]\n", 10);</code>	[ 10] 5桁で表示。不足分は空白。
		<code>printf("[%05d]\n", 10);</code>	[00010] 5桁。不足はゼロで埋める。
%f	実数を表示	<code>printf("[%f]\n", 12.345);</code>	[12.345000] 桁数不定
		<code>printf("[%9.5f]\n", 12.345);</code>	[ 12.34500] 小数点含めて全体が 9桁、小数以下が 5桁。
%e	実数を表示	<code>printf("[%e]\n", 12.345);</code>	[1.234500e+01] 浮動小数点で表示

□ ループ for 文、while 文、do~while 文

```
for( 開始時処理 ; 繰り返し条件式 ; 繰り返し毎処理 ) {
    繰り返す処理
}
```

```
while( 繰り返し条件式 ) {
    繰り返す処理
}
```

```
do {
    繰り返す処理
} while( 繰り返し条件式 );
```

□ scanf 文

```
scanf("書式", &変数 1, &変数 2, ... &変数 n);
```

"書式"には入力する変数の数だけ % に続けて入力する形式を変換文字列で指定。  
%d が十進数の整数、%f は float 型の実数、%lf は double 型の実数に対応。

□ 条件式

演算子	意味	利用例
==	等しい	if (a==b) { ... }
!=	等しくない	if (a!=b) { ... }
>	左辺が大きい	if (a>b) { ... }
>=	左辺が等しいか大きい	if (a>=b) { ... }
<	左辺が小さい	if (a<b) { ... }
<=	左辺が等しいか小さい	if (a<=b) { ... }

演算子	意味	利用例
&&	AND (両方の条件が成立したら)	if (a==b && a<100) { ... } (a と b が等しく、かつ a が 100 未満なら真)
	OR (どちらかの条件が成立したら)	if (a==b    a<100) { ... } (a と b が等いか、または a が 100 未満なら真)
!	NOT (条件の反転)	if (! a==b) { ... } (a と b が等しくなければ真)

□ if 文

```
if ( 条件式 ) {  
    条件が成立したときの処理 ;  
}
```

```
if ( 条件式 ) {  
    条件が成立したときの処理 ;  
} else {  
    条件が成立しなかったときの処理 ;  
}
```

```
if(条件 1) {  
    条件 1 が成立したときの処理;  
} else if(条件 2) {  
    条件 2 が成立したときの処理;
```

中略： else if 文は幾つでも書ける

```
} else if(条件 n) {  
    条件 n が成立したときの処理;  
} else {  
    条件 1 ... n が成立しなかったときの処理;  
}
```

最後の else 節は省略可

□ switch 文

```
switch( 式 ) {  
case 定数 1:  
    処理 1;  
    break;  
case 定数 2:  
    処理 2;  
    break;  
.. (略) ..  
default:  
    処理 n;  
}
```

- ・ case に続く条件部分には定数または定数式のみ。
- ・ default は省略可能。(該当しないケースでは何もせずに switch を通過。)
- ・ 各処理の最後には break; が必要。