

基本的なプログラミング1

01 print関数:画面に表示する1

- 以下のプログラムを入力し実行

```
print('Hello World')
```

- 実行結果 Hello World

- 関数 print(データ)
 - データには数値・数式・文字・変数を入れる
 - 文字は ' (シングルコーテーション) でくくる
 - ' の間は日本語が利用できる

02 print関数:画面に表示する1-練習

- 関数 print(データ)
 - データには数値・数式・文字・変数を入れる
 - 文字は ' (シングルコーテーション) でくくる
 - 'の間は日本語が利用できる
- 練習
 1. こんにちは と日本語で表示するプログラム
 2. 自分の名前を表示するプログラム

実行結果

こんにちは
自分の名前

03 print関数:画面に表示する2

- 以下のプログラムを入力し実行

```
print(10)
```

- 実行結果 10
- 関数 print(データ)
 - 数値はそのまま入力できる

04 print関数:画面に表示する3

- 以下のプログラムを入力し実行

```
print(2**6)
```

- 実行結果 **64**
 - $2^{**}6$ は26のこと
- 関数 `print(データ)`
 - 数式を入力することもできる

計算	演算子	例	結果
足し算	+	3+5	8
引き算	-	3-5	-2
掛け算	*	3*5	15
割り算	/	3/5	0.6
商	//	5//3	1
余り	%	5%3	2
累乗	**	5**3	125

05 print関数:画面に表示する4-練習

- 関数 `print(データ)`
 - 数式を入力することもできる

- 練習

- 2の10乗を表示するプログラム
- 54321を17で割った余りを表示するプログラム

計算	演算子	例	結果
足し算	+	3+5	8
引き算	-	3-5	-2
掛け算	*	3*5	15
割り算	/	3/5	0.6
商	//	5//3	1
余り	%	5%3	2
累乗	**	5**3	125

実行結果

1024
6

06 print関数:画面に表示する5

- 以下のプログラムを入力し実行

```
print('No.', 3)
```

- 実行結果 No. 3

- 関数 print(データ)
 - 複数書くときは , で区切る

07 print関数:画面に表示する6-練習

- 関数 `print(データ)`
 - 複数書くときは , で区切る
- 練習
 1. HelloとWorldをつなげて表示するプログラム
 2. 第 3 回 をつなげて表示するプログラム

実行結果

```
Hello World  
第 3 回
```


08 変数を使う1

- 以下のプログラムを入力し実行

```
name='Taro'  
print(name)
```

- 実行結果 Taro

- 変数名=変数の値

- 変数に値を代入しておくと、変数名で値を呼び出せる
- 文字は ' (シングルコーテーション) でくくる

09 変数を使う2-練習

- 変数名=変数の値
 - 変数に値を代入しておくと、変数名で値を呼び出せる
 - 文字は ' (シングルコーテーション) でくくる
- 練習
 1. 変数nameに自分の名前を代入し、表示するプログラム
 2. 変数schoolに立川高校を代入し、表示するプログラム
 3. 変数nameとschoolを使って、立川高校 自分の名前と表示するプログラム

実行結果

```
自分の 名前  
立川高校  
立川高校 自分の 名前
```

10 input関数:文字を入力する1

- 以下のプログラムを入力し実行

```
name=input('Name?')  
print(name)
```

- 実行例

Name? **Hanako**
Hanako

自分の名前を
入力して[Enter]

- input(メッセージ)

- メッセージを表示して文字を受け取る
- 変数=input(メッセージ) として変数に代入する

11 input関数:文字を入力する2-練習

- input(メッセージ)
 - メッセージを表示して文字を受け取る
 - 変数=input(メッセージ) として変数に代入する
 - 受け取った値は変数に代入しないと消えてしまう

• 練習

天気は? と表示して文字を入力させ、
受け取った文字を変数weatherに代入し、
今日は ○○ ですね と表示するプログラム

実行結果

天気は? **晴れ**
今日は 晴れ ですね

入力して
[Enter]

12 input関数:文字を入力する3

- 以下のプログラムを入力し実行

```
moji=input('Number?')  
print(moji*3)
```

- 実行例

Number?5

5を入力して[Enter]

555

3倍にならない

- input(メッセージ)

- メッセージを表示して文字を受け取る
- 数を入力しても文字扱い

正しく動いたら
2, -6などの数や
文字を入力してみる

13 int関数:文字列の数を数値にする1

- 以下のプログラムを入力し実行

```
suu=int(input('Number'))  
print(suu*3)
```

- 実行例

Number?
15

5

5を
入力して[Enter]

- int(数を表す文字列)
 - 数を表す文字列を数値に変換する

正しく動いたら
2,15,-6などを
入力し
何度か実行する

14 int関数:文字列の数を数値にする2

- 以下のプログラムを入力し実行

```
moji='5'  
print(moji*3)  
suuti=int(moji)  
print(suuti*3)
```

- 実行例 555
 15

- int(数を表す文字列)

15 int関数:文字列の数を数値にする3-練習

- int(数を表す文字列)
 - 数を表す文字列を数値に変換する
 - 数値を入力させるなら int(input())

• 練習

1. 数を入力させ、5乗した結果を表示するプログラム

実行結果

Number?
3125

5

入力して
[Enter]

16 if文:条件によって違う処理をする1

- 以下のプログラムを入力し実行

```
num=int(input('age?'))  
if num<15:  
    print('child')  
else:  
    print('adult')
```

4つ半角スペース

文末に:(コロン)

- 実行例

age?
adult

15

15を
入力して[Enter]

何度か実行し
20,12,50等の整数を
入力して
[Enter]キーを押す

17 if文:条件によって違う処理をする2

- 条件分岐

if 条件式:

真の時の処理

else:

偽の時の処理

は半角スペース

- 文末の:(コロン)・文頭の4スペースが必須

- :を入力後[Enter]キーを押すと自動的にスペースが入る

18 if文:条件によって違う処理をする3-練習

• 条件分岐

if 条件式:

真の時の処理

else:

偽の時の処理

は半角スペース

• 練習

- 数を入力させ、5より大きければ1、そうでなければ0を表示する

実行結果

Number?

5

0

入力して
[Enter]

19 if文:条件によって違う処理をする4

- 以下のプログラムを入力し実行

```
moji=input('Number?')
num=int(moji)
if num%2==0:
    print('Even')
else:
    print('Odd')
```

4つ半角スペース

文末に:(コロン)

Evenは偶数
Oddは奇数

- 実行例

Number?
Odd

15

15を
入力して[Enter]

何度か実行し、
2,19,21等の整数を
入力して
[Enter]キーを押す

20 if文:条件によって違う処理をする5

- 比較演算子:条件式に使う

演算子	利用例	意味
<	$A < B$	AはBより小さい
<=	$A \leq B$	AはB以下
>	$A > B$	AはBより大きい
>=	$A \geq B$	AはB以上
==	$A == B$	AとBは等しい
!=	$A != B$	AとBは等しくない

21 if文:条件によって違う処理をする6

- 演算子: 計算式に使う

演算子	利用例	意味
+	$A+B$	$A+B$
-	$A-B$	$A-B$
*	$A*B$	$A\times B$
/	A/B	$A\div B$
//	$A//B$	$A\div B$ の商
%	$A\%B$	$A\div B$ の余り
**	$A**B$	A^B

22 if文:条件によって違う処理をする7-練習

• 条件分岐

if 条件式:

真の時の処理

else:

偽の時の処理

演算子	利用例	意味
<	A<B	AはBより小さい
<=	A<=B	AはB以下
>	A>B	AはBより大きい
>=	A>=B	AはB以上
==	A==B	AとBは等しい
!=	A!=B	AとBは等しくない

• 練習

- 数値を入力させ、175以上ならL、それ未満ならMと表示するプログラム

入力して
[Enter]

実行結果

Number?

158

M

23 for文:繰り返して実行する1

- 以下のプログラムを入力し実行

```
for i in range(5):  
    print('Hello!')
```

文末に:(コロン)

- 実行例

```
Hello!  
Hello!  
Hello!  
Hello!  
Hello!
```

4つ半角スペース

- 繰り返して実行する反復構造

24 for文:繰り返して実行する2

- for 変数 in range(回数) :

- ■ ■ ■ 繰り返したい処理

■ は半角スペース

- 文末の:(コロン)・文頭の4スペースが必須

- :を入力後[Enter]キーを押すと自動的にスペースが入る

25 for文:繰り返して実行する3

- 以下のプログラムを入力し実行

4つ半角スペース

```
for i in range(5):  
    print(i)
```

文末に:(コロン)

- 実行例

```
0  
1  
2  
3  
4
```

26 for文:繰り返して実行する4

- for i in range(回数) :
print(i)

i が範囲内の間と読める

- for文では変数に順に値が代入される
 - range(5)は 0,1,2,3,4 が代入される
 - プログラミングでは0から数えることが多い

27 for文:繰り返して実行する5-練習

- 0から回数まで表示するプログラム

```
for i in range(回数):  
    print(i)
```

ヒント
print関数の()は
計算式が書ける

- 練習

- 1~100までの数を表示するプログラム

実行結果

1
2
3
⋮
98
99
100

28 for文:繰り返して実行する6

- 以下のプログラムを入力し実行

```
s=0
for i in range(10):
    s=s+i
print(s)
```

4つ半角スペース

文末に:(コロン)

繰り返す

- 実行例

45

繰り返さない

0から9までの和

29 for文:繰り返して実行する7

- 変数と繰り返しの組み合わせ
- プログラムを日本語にすると

合計を入れる変数 s を用意して0を代入
for文で i を0から10未満にして繰り返し:
 s と i の和を s に代入
 s を表示する

```
s=0
for i in range(10):
    s=s+i
print(s)
```

30 for文:繰り返して実行する8

- プログラム動作中の i と s の変化

i	s
0	0
1	$0+1=1$
2	$1+2=3$
3	$3+3=6$
4	$6+4=10$
5	$10+5=15$
6	$15+6=21$
7	$21+7=28$
8	$28+8=36$
9	$36+9=45$

どのように変化するか、
自分で考えてみるとよい

```
s=0
for i in range(10):
    s=s+i
print(s)
```

45

27 for文:繰り返して実行する-練習

- 0から9までの和

```
s=0
```

```
for i in range(10):
```

```
    s=s+i
```

```
print(s)
```

- 練習

- 1~1000までの数の和を表示するプログラム

実行結果 500500