

# 基本的なプログラミング

# 学習の仕方

- プログラミング環境を用意
  - PC室ならIDLEを起動
  - Surface等なら
    - Googleコラボトリー(おすすめ)  
<https://colab.research.google.com>
    - paiza.io(ちょっと癖あり)  
<https://paiza.io/ia>
- できるまで先に進まない
  - 正しく直せば正しく動く
  - エラーメッセージをしっかりと読もう(検索してもよい)
  - エラーを修正するのがプログラミングの重要ポイント

# 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([F5]→[Enter])

プログラム1 `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

## 02 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([F5]→[Enter])

プログラム2 `print(10)`

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

## 03 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([F5]→[Enter])

プログラム3 `print(2**6)`

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

# 04 print関数:画面に表示する4

- 以下のプログラムを入力し実行([F5]→[Enter])

プログラム4 `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

# 05 変数を使う1

- 以下のプログラムを入力し実行([F5]→[Enter])

プログラム5 

```
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

## 06 変数を使う2

- 以下のプログラムを入力し実行([F5]→[Enter])

プログラム6

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

- 実行結果  
Taro  
Jiro
- 変数名=変数の値 と書いて変数に代入
  - 同じ変数に違う値を代入すると上書きされる



# 07 変数を使う

プログラムを  
後ろに2行  
追加する

- 練習問題: プログラム6に加筆
- 追加するプログラム
  - 変数 name に Saburo を代入する
  - nameを表示する

## • 実行例

Taro

Jiro

Saburo

# 08 input関数:文字を入力する

- 以下のプログラムを入力し実行([F5]→[Enter])

プログラム7 

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

- 実行例

Name? **Hanako**  
Hanako

自分の名前を  
入力して[Enter]

Paizaの時は  
画面下[入力]に  
あらかじめ入力

- 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

# 09 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([F5]→[Enter])

プログラム8

```
moji='25'  
print(moji)  
suuti=int(moji)  
print(suuti)
```

- 実行例

25

25

こちらは文字列(文字コードで記録)

こちらは数値(2進法で記録)

- int(数を表す文字列)

見た目は同じ

- 数を表す文字列を数値に変換する

# 10 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([F5]→[Enter])

```
プログラム9 moji=' 25'  
             suuti=int(moji)  
             print(suuti+3)  
             print(moji+3)
```

- 実行例

28

エラー

このエラーは直せないなので  
そのまま先に進む

- int(数を表す文字列)

- 数値にすると計算ができる
- intは英語のinteger(整数)に由来する

# 11 int関数:文字列の数を数値にする3

- 以下のプログラムを入力し実行([F5]→[Enter])

プログラム10 `moji=input('Number')`

`suuti=int(moji)`

`print(suuti**3)`

\*\* は累乗

- 実行例

Number?

5

125

5 を  
入力して[Enter]

Paizaの時は  
画面下[入力]に  
あらかじめ入力

正しく動いたら  
2,15,-6などを  
入力し  
何度か実行する

- 入力した数を計算ときはこの方法
  - 文字列として受け取り、数値に変換する

# 12 int関数: 文字列の数を数値にする

- 練習問題: プログラム10を改造
- 入力した数を5倍するプログラムを作る

×は \* で表す

## • 実行例

Number?  
25

5

5 を  
入力して[Enter]

Paizaの時は  
画面下[入力]に  
あらかじめ入力

# 13 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([F5]→[Enter])

プログラム11

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

4つ半角スペース

文末に:(コロン)

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

- 実行例

age?  
adult

15

15を  
入力して[Enter]

Paizaの時は  
画面下[入力]に  
あらかじめ入力

# 14 if文:条件によって違う処理をする1

- 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



# 15 if文:条件によって違う処理をする2

- 以下のプログラムを入力し実行([F5]→[Enter])

プログラム12

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

Evenは偶数  
Oddは奇数

文末に:(コロン)

4つ半角スペース

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

- 実行例

Number?  
Odd

15

15を  
入力して[Enter]

Paizaの時は  
画面下[入力]に  
あらかじめ入力

# 16 if文:条件によって違う処理をする2

- 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

# 17 if文:条件によって違う処理をする2

- 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
**	$A**B$	$A^B$

# 18 if文:条件によって違う処理をする

- 練習問題: プログラム12を改造
- 数値を入力させ、175以上ならLsize、それ未満ならMsizeと表示するプログラムを作る

## • 実行例

Number?  
Msize

158

158 を  
入力して[Enter]

Paizaの時は  
画面下[入力]に  
あらかじめ入力

# 19 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([F5]→[Enter])

プログラム13 

```
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

# 20 for文:繰り返して実行する1

- for 変数 in range(回数):

■ ■ ■ ■ 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

# 21 for文:繰り返して実行する2

- 以下のプログラムを入力し実行([F5]→[Enter])

プログラム14 

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

文末に:(コロン)

4つ半角スペース

- 実行例

0  
1  
2  
3  
4

## 22 for文:繰り返して実行する2

- 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



## 23 for文:繰り返して実行する

- 練習問題: プログラム14を改造
- 1~100までの数を表示するプログラムを作る

### • 実行例

1

2

3

4



98

99

100

ヒント  
print関数の( )は  
計算式が書ける

# 24 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([F5]→[Enter])

プログラム15

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

文末に:(コロン)

4つ半角スペース

繰り返す

- 実行例

45

繰り返さない

0から9までの和

# 25 for文:繰り返して実行する3

- 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)
```

## 26 for文:繰り返して実行する3

•  $i$  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

- 練習問題: プログラム15を改造
- 1~1000までの数の和を表示するプログラムを作る

- 実行例

500500

# 28 制御構造の組み合わせ1

- 以下のプログラムを入力し実行([F5]→[Enter])

プログラム16 `for i in range(100):`

4つ半角スペース

```
    num=i+1
```

8つ半角スペース

```
    if num%2==0:
```

```
        print(num)
```

文末に:(コロン)

- 実行例

2

4

98

100

100までの偶数

# 29 制御構造の組み合わせ1

- 反復構造と分岐構造の組み合わせ
- プログラムを日本語にすると
  - for文で  $i$  は0~100未満まで変化:
    - 変数numを用意して1~100まで変化するようにする
    - numを2で割った余りが0の時:
      - numを表示する

```
for i in range(100):  
    num=i+1  
    if num%2==0:  
        print(num)
```

# 30 制御構造の組み合わせ

- 練習問題: プログラム16を改造
- 1~100までの3の倍数を表示するプログラムを作る

## • 実行例

3

6



96

99



# 31 制御構造の組み合わせ2

- 以下のプログラムを入力し実行([F5]→[Enter])

プログラム17

```
s=0
for i in range(100):
    num=i+1
    if num%2==0:
        s=s+num
print(s)
```

4つ半角スペース

8つ半角スペース

文末に:(コロン)

- 実行例

2550

100までの偶数の  
和

## 32 制御構造の組み合わせ2

- プログラムを日本語にすると
  - 和を入れるための変数sを用意して0を代入
  - for文で i は0~100未満まで変化:
    - 変数numを用意して1~100まで変化するようにする
    - numが2で割り切れるとき:
      - s+numをsに代入
  - sを表示

```
s=0
for i in range(100):
    num=i+1
    if num%2==0:
        s=s+num
print(s)
```

# 33 制御構造の組み合わせ2

- $i \cdot \text{num} \cdot s$  の関係

$i$	$\text{num}$	$s$
0	1	0
1	2	$0+2=2$
2	3	2
3	4	$2+4=6$
4	5	6
5	6	$6+6=12$
6	7	12
7	8	$12+8=20$
8	9	20
9	10	$20+10=30$

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(100):
    num=i+1
    if num%2==0:
        s=s+num
print(s)
```

## 34 制御構造の組み合わせ

- 練習問題: プログラム17を改造
- 1~100までの3の倍数の和を表示するプログラムを作る

- 実行例

1683

# 35 入力に対応したプログラム

- 以下のプログラムを入力し実行([F5]→[Enter])

```
プログラム18 suu=int(input('Number?'))  
              for i in range(suu):  
                  num=i+1  
                  if num%2==0:  
                      print(num)
```

4つ半角スペース

8つ半角スペース

文末に:(コロン)

- 実行例

Number? 15

2

4

14

入力した数  
以下の偶数

## 36 入力に対応したプログラム

- input関数で得た数を整数に直し変数suuに代入
  - input関数とint関数を組み合わせて利用すると便利
- suu までの範囲について繰り返し

プログラム18 `suu=int(input('Number?'))`

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

```
    num=i+1
```

```
    if num%2==0:
```

```
        print(num)
```

- 入力した数に対応したプログラムが作れる

# 37 入力に対応したプログラム

- プログラムを日本語にすると
  - input関数で得た数を整数に直し変数suuに代入
  - for文で変数 i を0からsuu 未満まで繰り返し:
    - 変数numにi+1を代入
    - もしnumが2で割り切れたら:
      - numを表示

```
suu=int(input('Number?'))
for i in range(suu):
    num=i+1
    if num%2==0:
        print(num)
```

# 38 range関数の使い方

- 以下のプログラムを入力し実行([F5]→[Enter])

プログラム19 

```
for i in range(2, 10):  
    print(i)
```

- 実行例

```
2  
3  
~~~~~  
8  
9
```

- range(開始値,上限)とすると開始値を指定できる



# 39 制御構造の組み合わせ3

- 以下のプログラムを入力し実行([F5]→[Enter])

```
プログラム20 suu=int(input('Number?'))
              sosu=True
              for i in range(2, suu):
                  if suu%i==0:
                      sosu=False
              if sosu==True:
                  print(suu, 'は素数')
              else:
                  print(suu, 'は素数ではない')
```

# 40 制御構造の組み合わせ3

## • 実行例

プログラム20

```
suu=int(input('Number?'))
sosu=True
for i in range(2, suu):
    if suu%i==0:
        sosu=False
if sosu==True:
    print(suu, 'は素数')
else:
    print(suu, 'は素数ではない')
```

数値を変え  
何回か実行し、  
正しい動作か  
確認する

Number?

15 は素数ではない

# 41 制御構造の組み合わせ

- 日本語にすると
  - 変数 `suu` に入力された数を代入
  - 変数 `sosu` に `True`(真)を代入
  - `i`を2から`suu`未満まで変えて繰り返し:
    - `suu`を`i`で割った余りが0なら:
      - 変数 `sosu` に`False`(偽)を代入
  - `sosu`が真なら:
    - `suu`は素数と表示
  - そうでなければ:
    - `suu`は素数でないと表示

```
suu=int(input('Number?'))
sosu=True
for i in range(2, suu):
    if suu%i==0:
        sosu=False
if sosu==True:
    print(suu, 'は素数')
else:
    print(suu, 'は素数ではない')
```

# 42 制御構造の組み合わせ

## • 練習問題1

- 入力された数までの素数を表示するプログラムを作る
  - ヒント: for文を2回使う

## • 練習問題2

- 入力された数に最も近い最大の(または最小の)素数を表示するプログラムを作る

作りたいプログラムを日本語にすると

- 変数 `suu` に入力された数を代入
- `j`を2から`suu`未満まで繰り返し:
  - 変数 `sosu` に True(真)を代入
  - `i`を2から`j`未満まで繰り返し:
    - `j`を`i`で割った余りが0なら:
      - 変数 `sosu` に False(偽)を代入
  - `sosu`が真なら:
    - `j`は素数と表示