

Python入門 第1回

目次

- [目次](#)
- [1. 整数 \(int\)](#)
- [2. 長整数 \(long\)](#)
- [3. 浮動小数点数 \(float\)](#)
- [4. 虚数 \(complex\)](#)
- [5. 論理値 \(bool\)](#)
- [6. 値無し \(None\)](#)
- [7. 文字列 \(str\)](#)
- [目次 \(続き\)](#)
- [9. 配列 \(リスト\)](#)
- [10. タプル](#)
- [11. 辞書 \(ディクショナリ\)](#)
- [12. 集合 \(セット\)](#)
- [目次 \(続き\)](#)
- [13. 条件分岐 \(if文\)](#)
- [14. 繰り返し処理 \(forループ\)](#)
- [15. 繰り返し処理 \(whileループ\)](#)
- [16. 関数定義](#)
- [17. クラス定義](#)

目次

1. 整数 (int)
 2. 長整数 (long)
 3. 浮動小数点数 (float)
 4. 虚数 (complex)
 5. 論理値 (bool)
 6. 値無し (None)
 7. 文字列 (str)
-

1. 整数 (int)

```
num = 1234 # 正の整数 (10進数)
```

2進数, 8進数, 16進数:

```
num = 0b11000100 # 2進数 (0b/0Bで始まる数値)  
num = 0o777      # 8進数 (0o/0Oで始まる数値)  
num = 0xffff      # 16進数 (0x/0Xで始まる数値)
```

2. 長整数 (long)

Python 2のみ:

```
num = 9223372036854775808L # L を末尾につける
```

3. 浮動小数点数 (float)

```
num = 1.234      # 浮動小数点数  
num = 1.2e3      # 指数表記 1.2 × 103  
num = 1.2E-3     # 指数表記 1.2 × 10-3
```

4. 虚数 (complex)

```
num = 3.14j
```

5. 論理値 (bool)

```
bool = True  
bool = False
```

6. 値無し (None)

```
x = None
```

7. 文字列 (str)

```
str = "Hello world"  
str = 'Hello world'
```

エスケープシーケンス:

```
str = "We can use \" in the string."  
str = 'We can use \' in the string.'
```

Raw文字列:

```
str = r'aaa\nbbb' # \n はバックスラッシュ文字(\)と小文字nとみなされる
```

目次 (続き)

- 9. 配列 (リスト)
- 10. タプル

11. 辞書 (ディクショナリ)

12. 集合 (セット)

9. 配列 (リスト)

```
my_list = [1, 2, 3, 4, 5]
my_list = [1, "a", "bbbb", True, 3.14, ["listのネストもできる", 516171]]
```

10. タプル

```
my_tuple = (1, 2, 3, 4, 5)
```

11. 辞書 (ディクショナリ)

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 7}
```

12. 集合 (セット)

```
my_set = {1, 2, 3, 4, 5}
```

目次 (続き)

13. 条件分岐 (if文)

14. 繰り返し処理 (forループ)

15. 繰り返し処理 (whileループ)

16. 関数定義

17. クラス定義

13. 条件分岐 (if文)

```
``python
x = 10
if x > 5:
    print("xは5より大きい")
elif x == 5:
    print("xは5です")
else:
    print("xは5より小さい")
```

14. 繰り返し処理 (forループ)

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

リストを使った繰り返し:

```
my_list = [1, 2, 3, 4, 5]
for item in my_list:
    print(item)
```

15. 繰り返し処理 (whileループ)

```
i = 0
while i < 5:
    print(i)
    i += 1
```

16. 関数定義

```
def my_function(x, y):  
    return x + y  
  
result = my_function(3, 4)  
print(result)
```

17. クラス定義

```
class MyClass:  
    def __init__(self, x, y):  
        self.x = x  
        self.y = y  
  
    def add(self):  
        return self.x + self.y  
  
my_instance = MyClass(3, 4)  
result = my_instance.add()  
print(result)
```