情報Ⅰ

4-1　　Python作法1
4-1-2　テキスト型
4-1-3　ブール型
4-1-4　型変換

4-1-2　テキスト型（str）･･･string　# 文字を扱う

- print('love')　# 値
- print('love'+'&'+'Peace')　# +でつなぐ
- print('love','&','Peace')　# ,でつなぐ
- print('every','¥n','day')　# 改行

```
print('love')
print('love'+'&'+'Peace')
print('love','&','Peace')
print('every','¥n','day')
```

```
love
love&Peace
love & Peace
every
 day
```

- t='world wide web'　　# 文字列定義
  print(t)　　# 文字表示
  print(len(t))　　# 文字数表示の内蔵関数
- print("day")　　# 文　字表示は‘～’か“～”
  print(5)　　# 順次構造により改行表示
- print('day',end ='')
  print(5)　　# end = ''で改行なし

```
t='world wide web'
print(t)
print(len(t))
```

```
print("day")
print(5)
print('day',end ='')
print(5)
```

4-1-3　ブール型（bool）　# true、falseの二択

- print(1>0.6)
- print(1<0.6)
- print(1==0.6)　　# 1と0.6は等しい（等号：==）
- print(1>=0.6)
- print(8!=2)　　# 1と0.6は等しくない（不等号：!=）

ポイント➡後に学ぶ分岐とは異なる

```
print(1>0.6)
print(1<0.6)
print(1==0.6)
print(8!=2)
```

```
True
False
False
True
```

4-1-4　型変換

pythonでは値を入力した時点で自動的に「型」を振り分けている･･･
（他の言語では変数定義で型を指定）

- x=10　　# 10を数値として扱う
  print(x+4)
- x='10'　　# 10を文字として扱う
  ここで数値のxは文字で上書き
- print(x+4)　　# 禁則(文字と数値は計算不可能)

```
x=10
print(x+4)
x='10'
print(x+4)
```

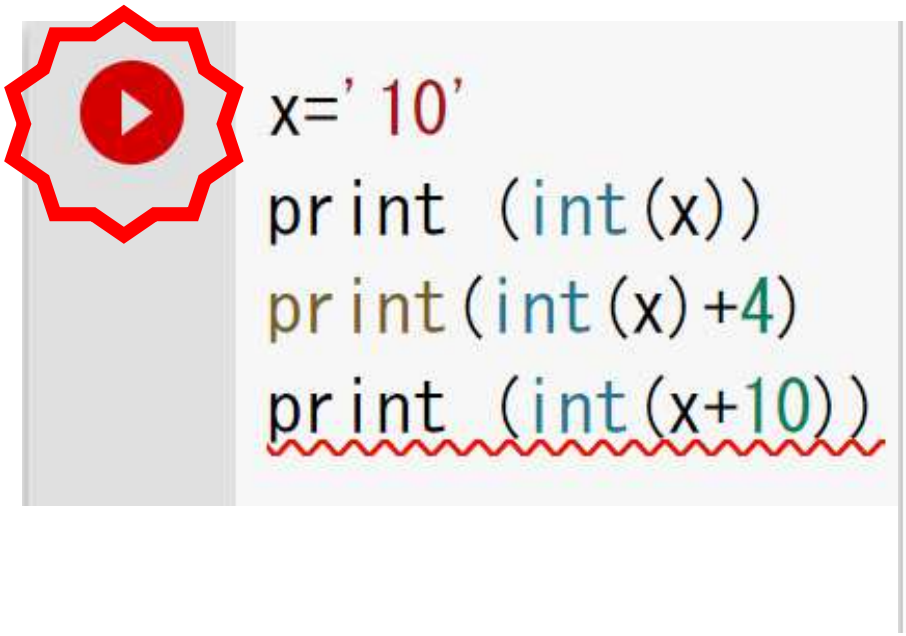
```
14
```

```
TypeError                                 Traceback (mo
<ipython-input-12-a0326a887150> in <module>
      2 print(x+4)
      3 x='10'
----> 4 print(x+4)

TypeError: can only concatenate str (not "int") to str
```

```
x='10'                  # 10を文字として扱う
print (int(x))          # 文字10を整数として表記
print(int(x)+4)         # 禁則回避(文字を数値に変換)
print (int(x+4))        # 禁則回避不可
```

ポイント➡後述のinput関数を使う場合など注意




```
10
14
---------------------------------------
TypeError
<ipython-input-14-76c37fa0ad0a>
      2 print (int(x))
      3 print(int(x)+4)
----> 4 print (int(x+10))
```