

## 例 2.1 变量的赋值和对象的地址演示

---

```
1 a=b=10 #a/b 指向同一个对象，地址应该一样
2 c=10    #c 的地址和 a/b 的一样吗？答案：一样
3 d=e=257 #d/e 指向同一个对象，地址应该一样
4 f=257   #f 的地址和 d/e 的一样吗？答案：不一样
5 m="python"
6 n="python" #m 的地址和 n 的一样吗？答案：一样
7 print(id(a),id(b),id(c))
8 print(id(d),id(e),id(f))
9 print(id(m),id(n))
```

---

输出结果（不同机器或不同运行结果不同）：

```
140707420518464 140707420518464 140707420518464
2075124619472 2075124619472 2075124619760
2075093509936 2075093509936
```

## 例 2.2 变量的赋值和对象的“相等”

---

```
# is 和 is not, 身份运算符 id
a = 'python'
b = 'python'
if a is b:
    print('a 和 b 指向同一个存储对象，有相同的标识')
else:
    print('a 和 b 没指向同一个存储对象，没有相同的标识')
if id(a) == id(b):
    print('a 和 b 指向同一个存储对象，有相同的标识')
else:
    print('a 和 b 没指向同一个存储对象，没有相同的标识')
b = 'b' #修改变量 b 的值
if a is b:
    print('a 和 b 指向同一个存储对象，有相同的标识')
else:
    print('a 和 b 没指向同一个存储对象，没有相同的标识')
if a is not b:
    print('a 和 b 没指向同一个存储对象，没有相同的标识')
else:
    print('a 和 b 指向同一个存储对象，有相同的标识')
```

---

## 例 2.3 表达式演示

---

```
1 a=b=c=90 #连续赋值
```

---

---

```
2 a,b,c=90,200,300 #拆包式赋值，表达式分别为 a、b、c 赋值 100、200、300。
3 if a<b<c: #条件满足，打印下面语句
4     print("python 可以使用连续比较表达式，就像数学中的一样。")
```

---

## 例 2.4 range 函数的使用

---

```
1 a=range(10)
2 b=range(10,20)
3 c=range(-10,10,2)
4 print(type(a)) #结果为: <class 'range'>
5 print(a.start,a.stop,a.step) #结果为: 0 10 1
6 print(b) #结果为: range(10, 20)
7 print(*c) #结果为: -10 -8 -6 -4 -2 0 2 4 6 8
8 print(list(b)) #结果为: [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
9 print(tuple(c)) #结果为: (-10, -8, -6, -4, -2, 0, 2, 4, 6, 8)
10 #下面举一个求 1 到 100 和的例子（先观察 range 函数的使用，后续讲 for 循环再深入）
11 s=0
12 for i in range(1,101):
13     s+=i
14 print(s) # 5050
```

---

## 例 2.5 math 包的使用

---

```
1 import math #这种导入方式，后续必须使用"math.函数名或常量名"形式，否则报错。
2 print("math.pi",math.pi) #结果: 3.141592653589793
3 print("math.floor(3.2)",math.floor(3.2))# 向下取整，结果: 3
4 print("math.ceil(4.5)",math.ceil(4.5)) # 向上取整，结果: 5
5 print("math.pow(3,4)",math.pow(3,4)) # =3**4，结果: 81.0

1 from math import * #这种导入方式，后续可以直接使用函数名，可以不用"math."
2 r=modf(3.14)
3 print("math.modf(3.14)",r)# 拆分小数和整数，结果: (0.14000000000000012, 3.0)
4 m,n=modf(3.14)
5 print(f"math.modf(3.14)的整数是 {m},小数是 {n}")
6 print("math.copysign(3,-5)",copysign(3,-5))#-5 的符号赋给 3，结果: -3.0
```

---

## 例 2.6 浮点数的相等比较

---

```
1 a, b, c=0.1, 0.2, 0.3 #拆包赋值
2 print(a+b) #结果为: 0.30000000000000004
3 print(a+b==c) #结果为: False
4 print(math.fabs(a+b-c)<1e-6) #结果为: True
5 print(math.isclose(a+b,c)) #结果为: True
6 print(math.isclose(a+b-c,0)) #结果为: False，该函数第二个参数最好不要为 0
```

---