Python-数据类型1

主讲人：班轩

Python-初体验

```
#猜数字游戏
import random


secret=random.randint(1,100)    #返回1和100之间的任意整数
print('猜数字游戏！\
我想了一个1—100之间的整数，你最多可以猜6次，\
看你可以猜出来吗？\n')       #多行注释也可以用用三个单引号 ''' 或者三个双引号 """ 将注释括起来
tries=1
p=0
while tries<=6:
    guess=int(input("1-100的整数，第%d次猜，请输入："%(tries,)))
    if guess ==secret:
        print("恭喜你猜对了，你只猜%d次！\n就是这个：%d!"%(tries,secret))
        p=1
        break
    elif guess>secret:
        print("不好意思，你猜的数大了一些！")
    else:
        print("不好意思，你猜的数小了一些！")
    tries+=1

if p==0:
    print("不好意思，你没猜中,下次再试试！")
```

注意对齐和缩进

注意字母大小写、空格

注意左右括号的配对

```
#输入某年某月某日，判断这一天是这一年的第几天？
import datetime

dtstr=input('Enter the datetime:(20200912):')        #用户输入的日期是字符串（str）。
dt=datetime.datetime.strptime(dtstr,"%Y%m%d")         #将输入的日期从字符串（str）转换为datetime。
another_dtstr=dtstr[:4]+'0101'                        #将用户输入的日期变成（XXXX0101）。
print(another_dtstr)
another_dt=datetime.datetime.strptime(another_dtstr,"%Y%m%d")
print(dtstr+"是当年的第"+str(int((dt-another_dt).days)+1)+"天")
```

Python-数据类型

数据类型		表示
数字	int(整数)	9
	float(浮点数)	类型3.14
	complex(复数)	python内置数据类型
布尔型	bool	True和False两种值
字符串	Str	用' '或" "或''' '''表示
列表	List	用[ ]符号表示
元组	Tuple	用( )符号表示
字典	Dict	用{ }符号表示

Python-数据类型

一、整数类型：int

最大的特点是：不限制大小

无论多复杂的算式都可以直接得到结果，比如2**100

1、算术运算符

a,b=10,5

运算符	实例
+	a + b 输出结果为 15
-	a - b 输出结果为 5
*	a * b 输出结果为 50
/	a / b 输出结果为 2
%	b % a 输出结果为 0
**	a**b 为 10 的 5 次方，输出结果为 100000
//	9//2 输出结果为 4，9.0//2.0 输出结果为 4.0

2、比较（关系）运算符 a,b=10,5

运算符	描述	实例
==	等于：比较对象是否相等	(a == b) 返回 False。
!=	不等于：比较两个对象是否不相等	(a != b) 返回 True.
>	大于：返回 x 是否大于 y	(a > b) 返回 False。
<	小于：返回 x 是否小于 y。所有比较运算符返回假，与特殊的变量 True 和 False 等价。注意大写	(a < b) 返回 True。
>=	大于等于：返回 x 是否大于等于 y。	(a >= b) 返回 False。
<=	小于等于：返回 x 是否小于等于 y。	(a <= b) 返回 True。

可以连续进行比较判断

```
>>> 7>3>=2
True
```

```
>>> 12<23<15
False
```

Python-数据类型

十进制	0	1	2	3	4	5	6	7	8
二进制	0	1	10	11	100	101	110	111	1000
八进制	0	1	2	3	4	5	6	7	10
十六进制	0	1	2	3	4	5	6	7	8
十进制	9	10	11	12	13	14	15	16	17
二进制	1001	1010	1011	1100	1101	1110	1111	10000	10001
八进制	11	12	13	14	15	16	17	20	21
十六进制	9	a	b	c	d	e	f	10	11

Python语言中可以直接用二进制、八进制和十六进制来表示整数，只要加一个 前缀用以标识几进制就可以。

进制	表示	例子	函数	
十进制(decimal)	无前缀数字	367		
二进制(binary)	**0b**前缀	0b101101111	**bin()**	**将10进制整数转换成 2 进制,以字符串形式表示。**
八进制(octal)	**0o**前缀	0o557	**oct()**	**将10进制整数转换成 8 进制,以字符串形式表示。**
十六进制(hexadecimal)	**0x**前缀	0x16f	**hex()**	**将10进制整数转换成16进制，以字符串形式表示。**

Python-数据类型

二、浮点数类型：float

操作与整数类似

浮点数受到17位有效数字的限制

特点：科学记数法

有效位数

特性：进制转换导致精度误差

```
>>> 355/113
3.1415929203539825
>>> 3.14159265358979323846
3.141592653589793
```

```
>>> 0.00000324003423423233240
3.2400342342323324e-06
>>> 213142332879382 7528.034
2.1314233287938276e+18
```

```
>>> 4.2+2.1==6.3
False
>>> 4.2+2.1
6.300000000000001
>>>
```

round() 方法返回浮点数x的四舍五入值。

```
>>> a=5.026
>>> b=5.000
>>> round(a, 2)
5.03
>>> round(b, 2)
5.0
```

```
>>> '%.2f'%a
'5.03'
>>> '%.2f'%b
'5.00'
```

Python-数据类型

三、逻辑数类型：bool

逻辑值仅包括真(True)/假(False)两个

用来配合if/while等语句做条件判断

3、逻辑运算

基本运算	运算符	表达式	返回值
非	not	not(3>2)	False
与	and	(25>=10)and (25<=100)	True
或	or	(25>100)or(25<2)	False

and和or是双目运算，由两个逻辑类型真值进行运算

not是单目运算，作用于一个逻辑类真值

优先级：not最高，and次之，or最低

Python-数据类型

4、各种类型对应的真值

- **整数、浮点数和复数类型**
  - 0是“假”，所有非0的数值都是“真”
- **字符串类型**
  - 空串("")是“假”，所有非空串都是“真”
- **所有序列类型（包括字符串）**
  - 空序列是“假”，所有非空的序列都是“真”
- **空值None**
  - 表示“无意义”或“不知道”，也是“假”

Python-数据类型

运算符说明	Python运算符	优先级	结合性	优先级顺序
小括号	( )	19	无	高
乘方	**	16	左	^
按位取反	~	15	右	|
符号运算符	-（负号）	14	右	|
乘除	*、/、//、%	13	左	|
加减	+、-	12	左	|
比较运算符	==、!=、>、>=、<、<=	7	左	|
逻辑非	not	4	右	|
逻辑与	and	3	左	|
逻辑或	or	2	左	低

Python-数据类型

四、字符串类型：str

字符串就是把一个个**文字的字符**“串起来”的数据

文字字符包含有拉丁字母、数字、标点符号、特殊符号，以及各种语言文字自字符。

表示字符串数字

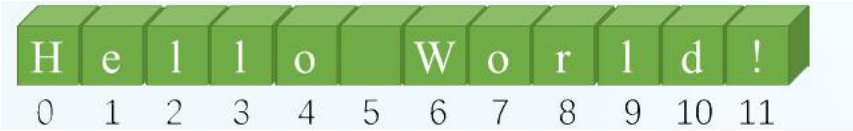
1、用双引号或者单引号都可以表示字符串，但必须成对

2、多行字符串用三个连续单引号（或双引号）表示

```
>>> "abc"
'abc'
>>> 'abc'
'abc'
>>> '''abc def
ghi jk'''
'abc def\nghi jk'
>>> """abc def
ghi jk"""
'abc def\nghi jk'
>>>
```

Python-数据类型

“Hello World!”



“你好，世界！”

Python-数据类型

常见的字符串操作

1、获取字符串的长度：len()函数

```
>>> len('我们')
2
>>> len("abcdef")
6
>>> a='Hello World!'
>>> len(a)
12
```

2、[ ]通过索引获取字符串中字符

```
>>> a='Hello World!'
>>> len(a)
12
>>> a[1]
'e'
>>>
```

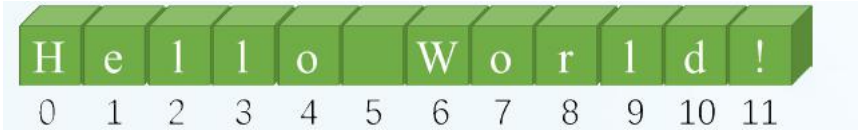
[start : end ：step]截取字符串中的一部分

```
>>> a='Hello World!'
>>> a[2:8:2]
'loW'
>>> a[2:8]
'llo Wo'
>>>
```

start是闭区间，end是开区间

[start:end)

“Hello World!”



Python-数据类型

常见的字符串操作

3、+:将两个字符进行连接，得到新的字符串。

4、*：将字符串重复若干次，生成新的字符串。

5、==：判断字符串内容是否相等

6、in：判断字符串中是否包含某个字符串

not in :判断字符串中是否不包含某个字符串

```
>>> a='Hello World!'
>>> b='abc'
>>> a+b
'Hello World!abc'
>>> a*3
'Hello World!Hello World!Hello World!'
>>> a=='Hello World!'
True
>>> 'H' in a
True
>>> 'Hello' in a
True
>>> 'p' not in a
True
>>>
```

Python-数据类型

ord() 函数它以一个字符（长度为1的字符串）作为参数，返回对应的 ASCII 数值或者 Unicode 数值，如果所给的 Unicode 字符超出了你的 Python 定义范围，则会引发一个 TypeError 的异常。

```
>>>ord('a')
97
>>> ord('b')
98
>>> ord('c')
99
```

chr() 用一个范围在 range（256）内的（就是0～255）整数(可以是10进制也可以是16进制的形式的数字)作参数，返回值是当前整数对应的 ASCII 字符.

```
>>>print chr(0x30), chr(0x31), chr(0x61)   # 十六进制
0 1 a
>>> print chr(48), chr(49), chr(97)        # 十进制
0 1 a
```

Python-数据类型

a 值为 "Hello"， b 为 "Python"

操作符	实例
+	>>>a + b 'HelloPython'
*	>>>a * 2 'HelloHello'
[]	>>>a[1] 'e'
[:]	>>>a[1:4] 'ell'
in	>>>"H" in a True
not in	>>>"M" not in a True
r/R	>>> print r'\n' \n >>> print R'\n' \n

Python-数据类型

对字符串的高级应用

1、split() 通过指定分隔符对字符串进行切片，

2、join() 方法用于将序列中的元素以指定的字符连接生成一个新的字符串。

```
>>> 'Hello World! 123'.split(' ')
['Hello', 'World!', '123']
>>> '-'.join(['Hello','World!','123'])
'Hello-World!-123'
>>>
```

3、str.lower() 将字符串中所有大写字符转换为小写。

4、str.upper() 将字符串中的小写字母转换为大写字母。

5、str.swapcase() 方法用于对字符串的大小写字母进行转换。

```
>>> 'Hello world!'.lower()
'hello world!'
>>> 'Hello world!'.upper()
'HELLO WORLD!'
>>> 'Hello world!'.swapcase()
'hELLO WORLD!'
```

Python-数据类型

对字符串的高级应用

6、str.ljust() 方法返回一个原字符串**左对齐**,并使用空格填充至指定长度的新字符串。如果指定的长度小于原字符串的长度则返回原字符串。

7、str.rjust() 返回一个原字符串**右对齐**,并使用空格填充至长度 width 的新字符串。如果指定的长度小于字符串的长度则返回原字符串。

8、str.center() 方法返回一个指定的宽度 width **居中**的字符串，fillchar 为填充的字符，默认为空格。

```
>>> 'Hello World!'.ljust (30)
'Hello World!                  '
>>> 'Hello World!'.rjust (30)
'                  Hello World!'
>>> 'Hello World!'.center (30)
'         Hello World!         '
>>>
```

Python-数据类型

9、str.replace() 方法把字符串中的 old（旧字符串） 替换成 new(新字符串)，如果指定第三个参数max，则替换不超过 max 次。

str.replace(old, new[, max])

old -- 将被替换的子字符串。

new -- 新字符串，用于替换old子字符串。

max -- 可选字符串, 替换不超过 max 次

```
>>> 'IT is new,IT is power, IT is symbol'.replace('IT','science')
'science is new,science is power, science is symbol'
>>>
```

Python-数据类型

四、数据类型转换

float() 函数用于将整数和字符串转换成浮点数。

str() 函数将对象转化为适于人阅读的形式。

int() 函数用于将一个字符串或数字转换为整型。

int(x, base=10)

x -- 字符串或数字。

base -- 进制数，默认十进制。

bool() 函数用于将给定参数转换为布尔类型，如果没有参数，返回 False。

```
>>>float(1)
1.0
>>> float(112)
112.0
>>> float(-123.6)
-123.6
>>> float('123')     # 字符串
123.0
```

```
>>>s = 'RUNOOB'
>>> str(s)
'RUNOOB'
>>> dict = {'runoob': 'runoob.com', 'google': 'google.com'};
>>> str(dict)
"{'google': 'google.com', 'runoob': 'runoob.com'}"
```

```
>>>int()              # 不传入参数时，得到结果0
0
>>> int(3)
3
>>> int(3.6)
3
>>> int('12',16)       # 如果是带参数base的话，12要以字符串的形式进行输入，12 为 16进制
18
>>> int('0xa',16)
10
>>> int('10',8)
8
```

```
>>>bool()
False
>>> bool(0)
False
>>> bool(1)
True
>>> bool(2)
True
```

Python-初体验

给数据命名：

命名语法：

<名字>=<数据>

命名规则：

由字母、数字、下划线组成，不能以数字开头。字母区分大小写。

不能带特殊符号（如空格、标点、运算符等）

名字的第一个字符不能数字开头

Python-初体验

名字（name）

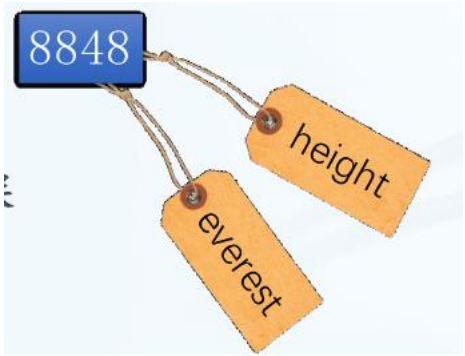
height=8848

名字像是一个标签，通过赋值来“贴”在某个数据数值上

名字和数值的关联称为**引用**。

关联数值后的名字，就拥有了数据的**值**（value）和**类型**（type）

一个数值可以和多个名字关联。

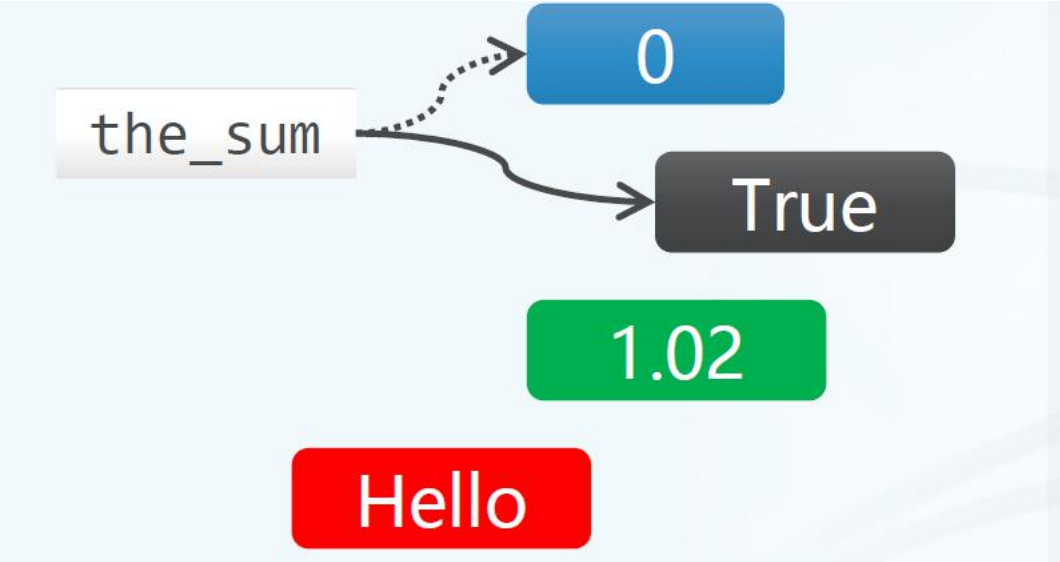


Python-初体验

与数值关联的名字也称作变量。 <变量>=<数据>

变量可以随时指向任何一个数据对象，比如True，1.02或者“Hello”

变量的类型随着指向的数据对象类型改变而改变。



Python-初体验

<名字/变量>=<数据>

名字与数值关联的过程，称为给变量赋值

“=”(赋值号)，计算等号右边式子的值，赋值给等号左边的变量。

合并赋值

a=b=c=1

按顺序依次赋值

a,b,c=7,8,9

简写赋值

运算符	描述	实例
=	简单的赋值运算符	c = a + b 将 a + b 的运算结果赋值为 c
+=	加法赋值运算符	c += a 等效于 c = c + a
-=	减法赋值运算符	c -= a 等效于 c = c - a
*=	乘法赋值运算符	c *= a 等效于 c = c * a
/=	除法赋值运算符	c /= a 等效于 c = c / a
%=	取模赋值运算符	c %= a 等效于 c = c % a
**=	幂赋值运算符	c **= a 等效于 c = c ** a
//=	取整除赋值运算符	c //= a 等效于 c = c // a

课堂作业：上机练习

1、数值的基本运算：33和7

+，-，*，/，//，%，**

hex(),oct(),bin()

2、类型转换：1，0，'abc'，1.2，False，''

str()，bool()，int()，str()

字符串基本操作：

+，*，len()，[]，in，ord()，chr()

含有中文的字符串

字符串的高级操作：

s='acdefg12345'

t='Mike and Tom'

感谢大家的聆听