

# Python笔记

注意python的缩进

## 变量和赋值

### 1. 变量命名规则

- 变量名必须是字母，数字，下划线，但是不能用数字开头（规则和C语言一样）
- 变量名大小写敏感
- 变量命名要做到“见名知意”

```
>>> name = 'william'
>>> print('hello,$s' %name)
hello,william

>>> age = 18
>>> height = 1800.2000001
>>> print('hello %s, 今年你%d岁了,有%3.2f' %(name,age,height))
hello william, 今年你18岁了,有180.20
```

### 2. 赋值

- 动态类型 type（变量名）
- python 不支持++ / --

## 字符串

- r 转义全部失效 `print(r"hello world! \n 'I love python'")`

## 切片

```
>>> word = 'abcdefghijklmn'
>>> print(word[0:3]) # 左闭右开
adb

# 字符串翻转
>>> a = 'abcdefg'
>>> print(a[::-1])
```

python 没有字符类型

## 注释

```
# 查看系统默认编码方式
>>> import sys
>>> print(sys.getdefaultencoding())
utf-8

"""
这也是以一种注释啦
"""
```

## 取余数

```
# 判断闰年
year = int(input("请输入某一年，将自动判断是否为闰年："))
if (year % 4) == 0:
    if (year % 100) == 0:
        if (year % 400) == 0:
            print("{0} 是闰年".format(year))
        else:
            print("{0} 不是闰年".format(year))
    else:
        print("{0} 是闰年".format(year))
else:
    print("{0} 不是闰年".format(year))

# 质数判断

num = int(input("请输入一个数字，来判断是否为质数："))

if num > 1:
    for i in range(2,num):
        if (num % i) == 0:
            print(num,"不是质数")
            break
    else:
        print(num,"是质数")
else:
    print("请输入一个大于1的整数")
```

## format 格式化函数

Python2.6 开始，新增了一种格式化字符串的函数 **str.format()**，它增强了字符串格式化的功能。

基本语法是通过 {} 和 : 来代替以前的 %。

format 函数可以接受不限个参数，位置可以不按顺序。

```
>>> "{} {}".format("hello", "world")    # 不设置指定位置，按默认顺序
'hello world'

>>> "{0} {1}".format("hello", "world")  # 设置指定位置
'hello world'

>>> "{1} {0} {1}".format("hello", "world") # 设置指定位置
'world hello world'

print("网站名: {name}, 地址 {url}".format(name="mengxun", url="mengxun.club"))

# 通过字典设置参数
>>> site = {"name": "mengxun", "url": "mengxun.club"}
>>> print("网站名: {name}, 地址 {url}".format(**site))
网站名: mengxun, 地址 mengxun.club

# 通过列表索引设置参数
>>> my_list = ['mengxun', 'mengxun.club']
>>> print("网站名: {0[0]}, 地址 {0[1]}".format(my_list)) # "0" 是必须的
网站名: mengxun, 地址 mengxun.club
```

## 数字格式化

下表展示了 `str.format()` 格式化数字的多种方法：

```
>>> print("{:.2f}".format(3.1415926))  
3.14
```

数字	格式	输出	描述
3.1415926	{:.2f}	3.14	保留小数点后两位
3.1415926	{:+.2f}	+3.14	带符号保留小数点后两位
-1	{:+.2f}	-1.00	带符号保留小数点后两位
2.71828	{:.0f}	3	不带小数
5	{:0>2d}	05	数字补零 (填充左边, 宽度为2)
5	{:x<4d}	5xxx	数字补 x (填充右边, 宽度为4)
10	{:x<4d}	10xx	数字补 x (填充右边, 宽度为4)
1000000	{:,}	1,000,000	以逗号分隔的数字格式
0.25	{:.2%}	25.00%	百分比格式
1000000000	{:.2e}	1.00e+09	指数记法
13	{:>10d}	13	右对齐 (默认, 宽度为10)

数字	格式	输出	描述
13	{:<10d}	13	左对齐 (宽度为10)
13	{:^10d}	13	中间对齐 (宽度为10)
11	<pre>'{:b}'.format(11)  '{:d}'.format(11) '{:o}'.format(11)  '{:x}'.format(11) '{:#x}'.format(11) '{:#X}'.format(11)</pre>	<pre>1011 11 13 b 0xb 0xB</pre>	进制

^, <, > 分别是居中、左对齐、右对齐，后面带宽度，:号后面带填充的字符，只能是一个字符，不指定则默认是用空格填充。

+ 表示在正数前显示+，负数前显示-；（空格）表示在正数前加空格

b、d、o、x 分别是二进制、十进制、八进制、十六进制

```
# 此外我们可以使用大括号 {} 来转义大括号，如下实例：
>>> print ("{} 对应的位置是 {}".format("mengxun"))
mengxun 对应的位置是 {}
```

## 整除

```
>>> print(5%3)
2
>>> print(5/-3)
-1.6666666666666667
>>> print(5// -3) # 取最小整数
-2
>>> print(5//3)
1
```

## 字符串比较

```
# 0~9 < A~Z < a~z
>>> print('az' < 'bc')
True

# 字符串只比较第一个值的大小

>>> print('123' < '23')
True
```

