

day18

今日内容

- 回顾和补充
- day16作业讲解

内容详细

1. Python入门

1.1 环境的搭建

mac系统上搭建python环境。

环境变量的作用：方便在命令行（终端）执行可执行程序，将可执行程序所在的目录添加到环境变量，那么以后无需再输入路径。

1.2 变量命名

- 变量

```
name = 'alex'
```

- 全局变量

```
NAME = "oldboy"
```

- 函数

```
def func_list():  
    pass
```

- 常量

不允许修改的值，Python中执行约定。

1.3 运算符

```
v = 1 or 2  
v = 0 or 2  
v = 1 and 2  
v = 0 and 2
```

```
val = v if v else 666
val = v or 666 # 源码
```

is和==的区别？

1.4 数据类型

- int
 - 整型
 - 其他进制转换十进制 `int('0b11011',base=2)`
 - int / long
 - 除法
- bool
 - 0 None 空
- str, 字符串类型, 一般用于内存中做数据操作。

```
v = 'alex' # unicode编码存储在内存。
```

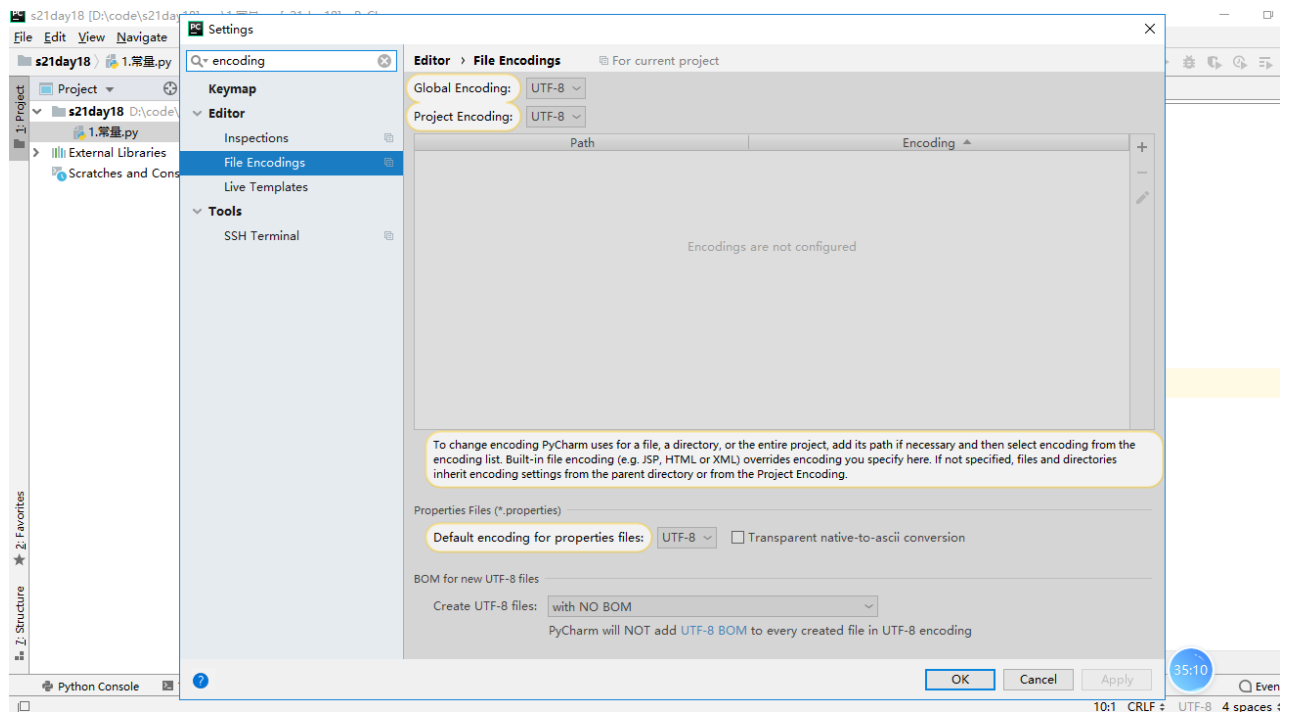
- bytes, 字节类型, 一般用于数据存储和网络传输。

```
v = 'alex'.encode('utf-8') # 将字符串转换成字节 (由unicode编码转换为utf-8编码)
v = 'alex'.encode('gbk')   # 将字符串转换成字节 (由unicode编码转换为gbk编码)
```

- list
- tuple
- dict
- set
- None

1.5 编码

- py默认解释器编码
 - py2
 - py3
- 以什么编码存储就要以什么编码打开 (建议Pycharm设置成UTF-8编码)。



- 常见的编码
 - ascii
 - unicode
 - utf-8
 - gbk
 - gb2312

1.6 深浅拷贝

针对可变类型，将其重新创建一份。

- 浅拷贝：第一层
- 深拷贝：所有层。

```
import copy

v = [11, 22, 33, [11, 22, 3]]
v1 = copy.copy(v1)
v2 = copy.deepcopy(v)
```

1.7 py2和py3区别

- 字符串类型不同
 - py3: str bytes
 - py2: unicode str
- 默认解释器编码
- 输入输出
- int
 - int long
 - 除法

- range和xrange
- 模块和包
- 字典
 - keys
 - py2: 列表
 - py3: 迭代器, 可以循环但不可以索引
 - values
 - py2: 列表
 - py3: 迭代器, 可以循环但不可以索引
 - items
 - py2: 列表
 - py3: 迭代器, 可以循环但不可以索引
- map/filter
 - py2: 返回列表
 - py3: 返回迭代器, 可以循环但不可以索引

2.函数

2.1 内置函数

- 常用内置函数: open / id / type / len / range ...
- is 和 == 的区别? 通过id来进行检查

2.2 自定义函数

- 函数式编程: 增加代码的可读性和重用性。
- 函数基本格式:

```
def show(name, age):
    """
    函数是干什么的...
    :param name:
    :param age:
    :return:
    """
    return None
```

- 函数做参数
- 函数做变量
- 函数做返回值
 - 闭包
 - 装饰器
- 生成器

```
def func():
    print(123)
    yield 1
    yield 3

# 生成器
v = func()
# 循环v时或v.__next__() 时。
```

```
def base():
    yield 88
    yield 99

def func():
    yield 1
    yield 2
    yield from base()
    yield 3

result = func()

for item in result:
    print(item)
```

生成器推导式

```
# def func():
#     result = []
#     for i in range(10):
#         result.append(i)
#     return result
# v1 = func()
v1 = [i for i in range(10)] # 列表推导式，立即循环创建所有元素。
print(v1)

# def func():
#     for i in range(10):
#         yield i
# v2 = func()
v2 = (i for i in range(10)) # 生成器推导式，创建了一个生成器，内部循环为执行。

# 面试题：请比较 [i for i in range(10)] 和 (i for i in range(10)) 的区别？
```

```
# 示例一
# def func():
#     result = []
#     for i in range(10):
#         result.append(i)
#     return result
# v1 = func()
# for item in v1:
```

```

#     print(item)

# 示例二
# def func():
#     for i in range(10):
#         def f():
#             return i
#         yield f
#
# v1 = func()
# for item in v1:
#     print(item())

# 示例三:
v1 = [i for i in range(10)] # 列表推导式, 立即循环创建所有元素。
v2 = (lambda :i for i in range(10))
for item in v2:
    print(item())

```

- 迭代器

3. 模块

3.1 内置模块

- 常见内置模块: json / datetime / time / os / sys

3.2 第三方模块

- requests
- xlrd

3.3 自定义模块

- 文件
- 文件夹 / 包

3.4 使用模块

- 导入
 - import 模块
 - from 模块.模块 import 模块
 - from 模块.模块.模块 import 函数
 - 相对导入【不推荐】

```

from . import xxx
from .. import xxx

```

注意: 文件和文件夹的命名不能是导入的模块名称相同, 否则就会直接在当前目录中查找。

- 调用模块内部元素

- 函数()
- 模块.函数()
- 主文件（运行的文件）

```
if __name__ == '__main__':  
    page.pager()
```

4.其他

- 推导式
- 三元运算
- lambda表达式

5.作业题讲解（day16）