

python基础第一天-基础知识

python基础第一天-基础知识

一、计算机基础知识

1. 【了解】 硬件系统
2. 【了解】 软件系统

二、认识Python

1. 【了解】 Python的介绍
2. PyCharm的使用
 - 2.1 【知道】 PyCharm创建工程
 - 2.2 【了解】 PyCharm设置字体

三、基础语法

1. 第一个Python程序
 - 1.1 【重点】 第一个Python程序
 - 1.2 【重点】 bug介绍
 - 1.3 【重点】 注释
 - 1.4 【了解】 PyCharm常用快捷键
2. 变量、类型、标示符
 - 2.1 【重点】 变量
 - 2.2 【重点】 类型
 - 2.3 【重点】 标示符命名规则
 - 2.4 【了解】 变量命名规范

2.5 【了解】 关键字

2.6 【了解】 如何进入Python交互模式

3. 输出和输入

3.1 【重点】 输出

3.2 【重点】 输入

3.3 【重点】 类型转换

4. 【知道】 算术、赋值、复合赋值运算符

四、学习方法

一、计算机基础知识

1. 【了解】 硬件系统

- 主机部分：中央处理器 + 内部存储器
- 外设部分：输入设备 + 输出设备 + 外部存储器

2. 【了解】 软件系统

- 系统软件：操作系统、编程程序
- 应用软件：音乐播放器、qq、wx、浏览器等

二、认识Python

1. 【了解】Python的介绍

1. Python 是一种编程语言

- 特点：解释型语言，需要解释器逐行解释执行。

2. Python 的优缺点

- 优点：简单易学
- 缺点：慢

3. Python 应用场景：**web**开发和**数据科学**

2. PyCharm的使用

2.1 【知道】PyCharm创建工程

- 根据讲义截图操作

2.2 【了解】PyCharm设置字体

- 根据讲义截图操作

三、基础语法

1. 第一个Python程序

1.1 【重点】 第一个Python程序

```
print("hello world!")
```

1.2 【重点】 bug介绍

- bug: 程序错误
- NameError: 名字错误
- SyntaxError: `print ("hello")` 语法错误
- TypeError: 类型错误
- IndentationError: 缩进错误

1.3 【重点】 注释

注释

- 对程序代码进行**解释说明**的文字，注释不是代码，不会执行
- 分类：单行注释、多行注释

```
# 单行注释
```

```
"""
```

```
多行注释1
```

```
"""
```

```
'''
```

```
多行注释
```

```
'''
```

1.4 【了解】 PyCharm常用快捷键

- 注释: `ctrl + /`
- 代码格式化: `ctrl + alt + l`
- 运行: `ctrl + shift + f10`

2. 变量、类型、标示符

2.1 【重点】 变量

- 变量作用：存储数据，类似于盒子、容器

```
# 变量名 = 数据
# 第一次赋值就是定义
num = 123
# 第二次赋值就是修改变量
num = 250
```

2.2 【重点】 类型

```
"""
250            int, 整型
3.14          float 浮点型
'mike'        字符串, 只要是' '格式的内容就是字符串
"mike"        字符串, 只要是" "格式的内容就是字符串
True, False   bool, 布尔, 计算True就是1,
False就是0
"""
```

- 定义变量不需要指定类型, 自动推导类型
- 查看类型: `type(变量名)`

```
a = 250
print(type(a))
```

2.3 【重点】 标示符命名规则

- 标示符：函数名、变量名
- 命名规则：
 - 数字，字母，`_`组成
 - 不能以数字开头
 - 不能和关键字重名
 - 建议不要和类型重名: `int`, `str`, ...
- 区分大小写
 - `Andy` != `andy`

2.4 【了解】 变量命名规范

- 规范: 见名知意
- Python变量名、函数名: `user_name`
- 驼峰命名：
 - 小驼峰: `userName`
 - 大驼峰: `UserName`

2.5 【了解】 关键字

- 关键字: python定义的有特定功能的标示符，起名不能和关键字重名

```
# 导入工具包
import keyword

# 打印关键字
print(keyword.kwlist)
```

2.6 【了解】 如何进入Python交互模式

- 按照讲义实现

3. 输出和输入

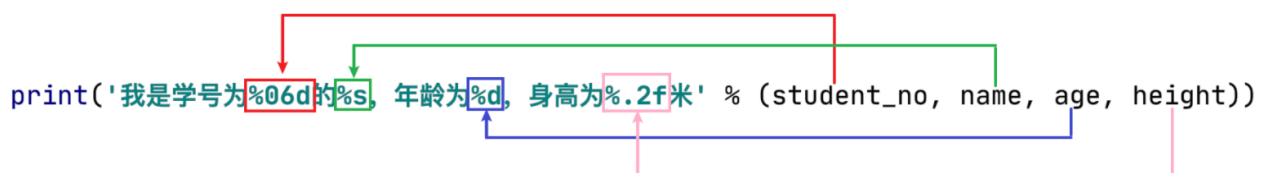
3.1 【重点】 输出

- 整数: `%d`
- 字符串: `%s`
- 小数: `%f`


```
student_no = 1
age = 18
name = "小明"
height = 1.78

# print("格式化字符串" % (变量1, 变量2, ...))
# print("格式化字符串" % 变量)

print("我是学号为%06d的%s, 年龄为%d, 身高为%.2f米" % (student_no, name, age, height))
```



```
print('我是学号为%06d的%s, 年龄为%d, 身高为%.2f米' % (student_no, name, age, height))
```

我是学号为000001的小明, 年龄为18, 身高为1.78米

3.2 【重点】 输入

- 输入后得到内容的类型是字符串
- 格式:

```
变量名 = input("提示信息: ")
```

- 看变量是什么类型, 通过 `type(变量名)` 来查看

3.3 【重点】 类型转换

- 类型不匹配，类型转换

```
int(x) # 把x转换成整形  
float(x) # 把x转换成浮点型  
str(x) # 把x转换成字符串
```

4. 【知道】 算术、赋值、复合赋值运算符

- 算术:

```
+, -, *, /, //, %, **
```

- 赋值:

```
=
```

- 复合赋值运算符

```
+=, -=, *=, /=, //=, %=, **=
```

```
a += 1 --> a=a+1
```

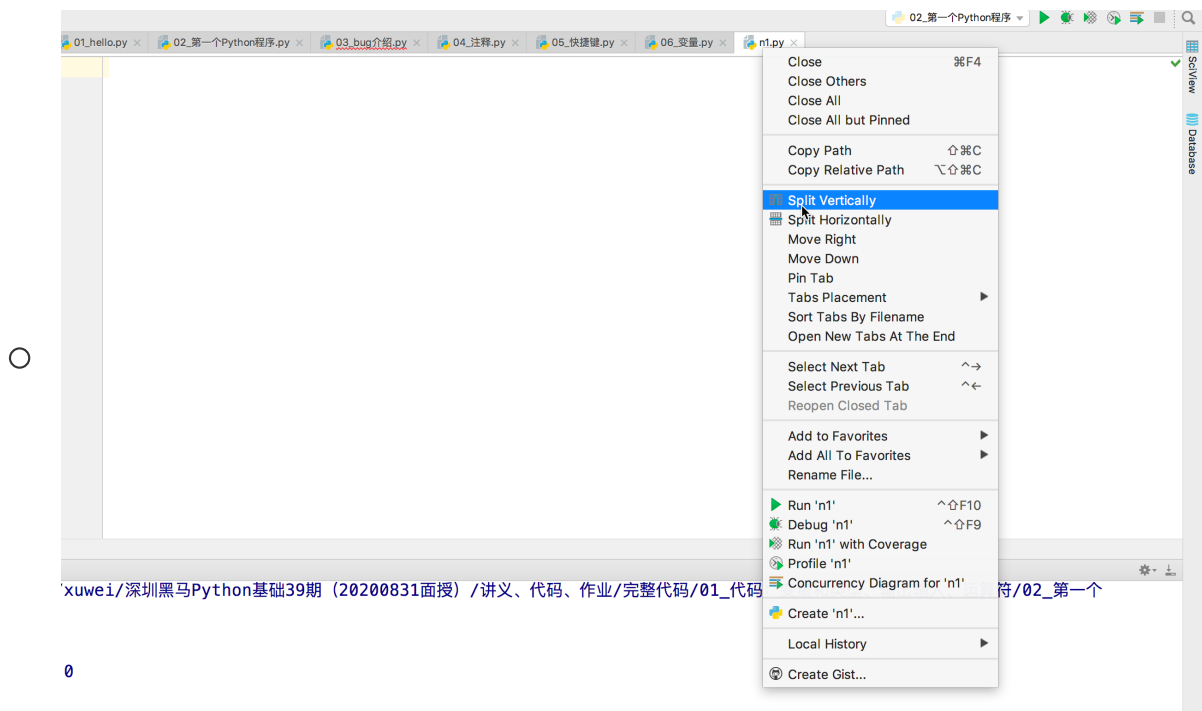
四、学习方法

1. 看笔记，看目录，回忆内容

- 如果有代码，回忆不起来，不要总想，没有太大意义

2. 代码学习：切忌光看不练

- 直接运行讲师写的代码，运行看效果：验证自己的想法和效果是否一致
- 屏蔽代码，抄代码，改代码：减少写错概率



- 重头写：练习

3. 做作业

