

多文件项目的演练

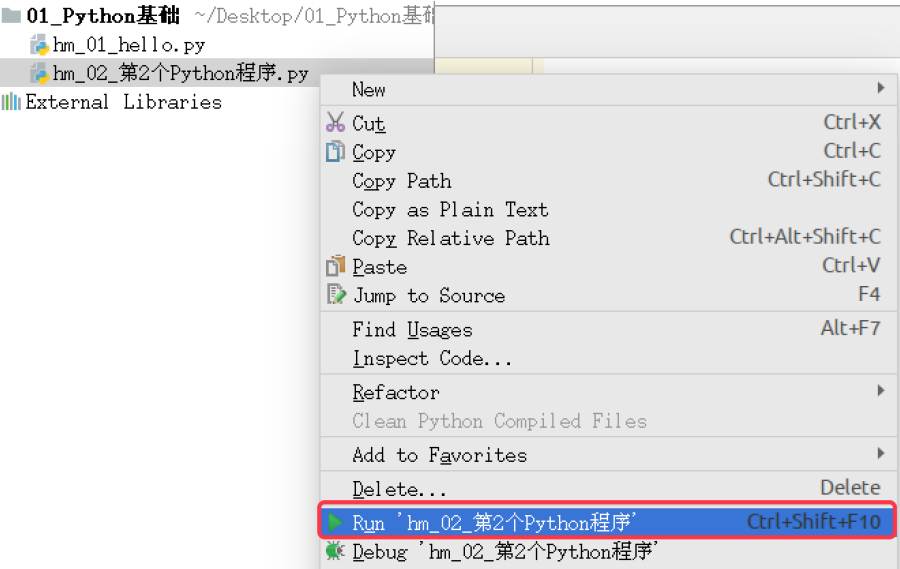
- 开发项目 就是开发一个专门解决一个复杂业务功能的软件
- 通常每一个项目 就具有一个独立专属的目录，用于保存所有和项目相关的文件
 - 一个项目通常会包含很多源文件

目标

- 在项目中添加多个文件，并且设置文件的执行

多文件项目演练

1. 在 01_Python基础 项目中新建一个 hm_02_第2个Python程序.py
2. 在 hm_02_第2个Python程序.py 文件中添加一句 print("hello")
3. 点击右键执行 hm_02_第2个Python程序.py



提示

- 在 PyCharm 中，要想让哪一个 Python 程序能够执行，必须首先通过鼠标右键的方式执行一下
- 对于初学者而言，在一个项目中设置多个程序可以执行，是非常方便的，可以方便对不同知识点的练习和测试
- 对于商业项目而言，通常在一个项目中，只有一个可以直接执行的 Python 源程序

远程管理常用命令

目标

- 关机/重启
 - shutdown
- 查看或配置网卡信息
 - ifconfig
 - ping
- 远程登录和复制文件
 - ssh
 - scp

01. 关机/重启

| 序号 | 命令 | 对应英文 | 作用 | |---| |---| |---| | 01 | shutdown 选项 时间 | shutdown | 关机 / 重新启动 |

1.1 shutdown

- shutdown 命令可以安全关闭或者重新启动系统

| 选项 | 含义 | |---| |---| |---| | -r | 重新启动 |

- 提示：
- 不指定选项和参数，默认表示 1 分钟之后关闭电脑
 - 远程维护服务器时，最好不要关闭系统，而应该重新启动系统

- 常用命令示例

```
'''bash
```

重新启动操作系统，其中 now 表示现在

```
$ shutdown -r now
```

立刻关机，其中 now 表示现在

```
$ shutdown now
```

系统在今天的 20:25 会关机

\$ shutdown 20:25

系统再过十分钟后自动关机

\$ shutdown +10

取消之前指定的关机计划

\$ shutdown -c ""

02. 查看或配置网卡信息

| 序号 | 命令 | 对应英文 | 作用 || --- | --- | --- || 01 | ifconfig | configure a network interface | 查看/配置计算机当前的网卡配置信息 || 02 | ping ip地址 | ping | 检测到目标 ip地址 的连接是否正常 |

2.1 网卡 和 IP 地址

网卡

- 网卡是一个专门负责网络通讯的硬件设备
- IP 地址是设置在网卡上的地址信息

我们可以把 电脑 比作 电话，网卡 相当于 SIM 卡，IP 地址 相当于 电话号码

IP 地址

- 每台联网的电脑上都有 IP 地址，是保证电脑之间正常通讯的重要设置

注意：每台电脑的 IP 地址不能相同，否则会出现 IP 地址冲突，并且没有办法正常通讯

提示：有关 IP 地址的详细内容，在就业班会详细讲解！

2.2 ifconfig

- ifconfig 可以查看 / 配置计算机当前的网卡配置信息

``bash

查看网卡配置信息

\$ ifconfig

查看网卡对应的 IP 地址

\$ ifconfig | grep inet ``

提示：一台计算机中有可能会有一个 物理网卡 和 多个虚拟网卡，在 Linux 中物理网卡的名字通常以 ensXX 表示

- 127.0.0.1 被称为 本地回环/环回地址，一般用来测试本机网卡是否正常

2.3 ping

``bash

检测到目标主机是否连接正常

\$ ping IP地址

检测本地网卡工作正常

\$ ping 127.0.0.1 ``

- ping 一般用于检测当前计算机到目标计算机之间的网络 是否通畅，数值越大，速度越慢

- ping 的工作原理与潜水艇的声纳相似，ping 这个命令就是取自 声纳 的声音
- 网络管理员之间也常将 ping 用作动词 —— ping 一下计算机X，看他是否开着

原理：网络上的机器都有 唯一确定的 IP 地址，我们给目标 IP 地址 发送一个数据包，对方就要返回一个数据包，根据返回的数据包以及时间，我们可以确定目标主机的存在

提示：在 Linux 中，想要终止一个终端程序的执行，绝大多数都可以使用 `CTRL + C`

03. 远程登录和复制文件

| 序号 | 命令 | 对应英文 | 作用 || --- | --- | --- || 01 | ssh 用户名@ip | secure shell | 关机 / 重新启动 || 02 | scp 用户名@ip:文件名或路径 用户名@ip:文件名或路径 | secure copy | 远程复制文件 |

3.1 ssh 基础（重点）

在 Linux 中 SSH 是 非常常用 的工具，通过 SSH 客户端 我们可以连接到运行了 SSH 服务器 的远程机器上

- SSH 客户端是一种使用 Secure Shell (SSH) 协议连接到远程计算机的软件程序
- SSH 是目前较可靠，专为远程登录会话和其他网络服务 提供安全性的协议
 - 利用 SSH 协议 可以有效防止远程管理过程中的信息泄露
 - 通过 SSH 协议 可以对所有传输的数据进行加密，也能够防止 DNS 欺骗和 IP 欺骗
- SSH 的另一项优点是传输的数据可以是经过压缩的，所以可以加快传输的速度

1) 域名 和 端口号

域名

- 由一串 用点分隔 的名字组成，例如：www.itcast.cn
- 是 IP 地址 的别名，方便用户记忆

端口号

- IP 地址：通过 IP 地址 找到网络上的 计算机
- 端口号：通过 端口号 可以找到 计算机上运行的应用程序
 - SSH 服务器 的默认端口号是 22，如果是默认端口号，在连接的时候，可以省略
- 常见服务端口号列表：

| 序号 | 服务 | 端口号 | --- | --- | --- | 01 | SSH 服务器 | 22 | 02 | Web 服务器 | 80 | 03 | HTTPS | 443 | 04 | FTP 服务器 | 21 |

提示：有关 端口号 的详细内容，在就业班会详细讲解！

2) SSH 客户端的简单使用

```
bash ssh [-p port] user@remote
```

- `user` 是在远程机器上的用户名，如果不指定的话默认为当前用户
- `remote` 是远程机器的地址，可以是 IP / 域名，或者是 后面会提到的别名
- `port` 是 SSH Server 监听的端口，如果不指定，就为默认值 22

提示：

- 使用 `wexit` 退出当前用户的登录

注意：

- `ssh` 这个终端命令只能在 `Linux` 或者 `UNIX` 系统下使用
- 如果在 `Windows` 系统中，可以安装 `Putty` 或者 `Xshell` 客户端软件即可

提示：

- 在工作中，SSH 服务器的端口号很有可能不是 22，如果遇到这种情况就需要使用 `-p` 选项，指定正确的端口号，否则无法正常连接到服务器

3) Windows 下 SSH 客户端的安装

- Putty <http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- Xshell <http://xshellcn.com>

建议从官方网站下载正式的安装程序

3.2 scp（掌握）

- `scp` 就是 `secure copy`，是一个在 Linux 下用来进行 远程拷贝文件 的命令
- 它的地址格式与 `ssh` 基本相同，需要注意的是，在指定端口时用的是大写的 `-P` 而不是小写的

```
_____
```

```
```bash
```

把本地当前目录下的 **01.py** 文件 复制到 远程 家目录下的 **Desktop/01.py**

注意：后面的路径如果不是绝对路径，则以用户的家目录作为参照路径

```
scp -P port 01.py user@remote:Desktop/01.py
```

把远程 家目录下的 **Desktop/01.py** 文件 复制到 本地当前目录下的 **01.py**

```
scp -P port user@remote:Desktop/01.py 01.py
```

加上 **-r** 选项可以传送文件夹

把当前目录下的 **demo** 文件夹 复制到 远程 家目录下的 **Desktop**

```
scp -r demo user@remote:Desktop
```

把远程 家目录下的 **Desktop** 复制到 当前目录下的 **demo** 文件夹

```
scp -r user@remote:Desktop demo ```
```

| 选项 | 含义 | --- | --- | | `-r` | 若给出的源文件是目录文件，则 `scp` 将递归复制该目录下的所有子目录和文件，目标文件必须为一个目录名 | | `-P` | 若远程 SSH 服务器的端口不是 22，需要使用大写字母 `-P` 选项指定端口 |

注意：

- `scp` 这个终端命令只能在 `Linux` 或者 `UNIX` 系统下使用
- 如果在 `Windows` 系统中，可以安装 `Putty`，使用 `pscp` 命令行工具或者安装 `FileZilla` 使用 `FTP` 进行文件传输

## FileZilla

- 官方网站：<https://www.filezilla.cn/download/client>
- FileZilla 在传输文件时，使用的是 `FTP` 服务 而不是 `SSH` 服务，因此端口号应该设置为 21

## 3.3 SSH 高级（知道）

- 免密码登录
- 配置别名

提示：有关 SSH 配置信息都保存在用户家目录下的 `~/.ssh` 目录下

### 1) 免密码登录

步骤

- 配置公钥
  - 执行 `ssh-keygen` 即可生成 SSH 钥匙，一路回车即可
- 上传公钥到服务器
  - 执行 `ssh-copy-id -p port user@remote`，可以让远程服务器记住我们的公钥

示意图

非对称加密算法

- 使用 公钥 加密的数据，需要使用 私钥 解密
- 使用 私钥 加密的数据，需要使用 公钥 解密

2)配置别名

每次都输入 `ssh -p port user@remote`，时间久了会觉得很麻烦，特别是当 `user`、`remote` 和 `port` 都得输入，而且还不好记忆

而 **配置别名** 可以让我们进一步偷懒，譬如用：`ssh mac` 来替代上面这么一长串，那么就在 `~/.ssh/config` 里面追加以下内容：

```
Host mac HostName ip地址 User itheima Port 22
```

保存之后，即可用 `ssh mac` 实现远程登录了，`scp` 同样可以使用

## 算数运算符

计算机，顾名思义就是负责进行 **数学计算** 并且 **存储计算结果** 的电子设备

### 目标

- 算术运算符的基本使用

#### 01. 算数运算符

- 算数运算符是 **运算符** 的一种
- 是完成基本的算术运算使用的符号，用来处理四则运算

| 运算符 | 描述 | 实例 || :--: | :--: | :--: || + | 加 | 10 + 20 = 30 || - | 减 | 10 - 20 = -10 || \* | 乘 | 10 \* 20 = 200 || / | 除 | 10 / 20 = 0.5 || // | 取整除 | 返回除法的整数部分（商） 9 // 2 输出结果 4 || % | 取余数 | 返回除法的余数 9 % 2 = 1 || \*\* | 幂 | 又称次方、乘方，2 \*\* 3 = 8 |

- 在 Python 中 `*` 运算符还可以用于字符串，计算结果就是字符串重复指定次数的结果

```
python In [1]: "-" * 50 Out[1]: '-----'
```

#### 02. 算数运算符的优先级

- 和数学中的运算符的优先级一致，在 Python 中进行数学计算时，同样也是：

- 先乘除后加减**
  - 同级运算符是 **从左至右** 计算
  - 可以使用 `()` 调整计算的优先级
- 以下表格的算数优先级由高到最低顺序排列

| 运算符 | 描述 || -- | :-- || \*\* | 幂 (最高优先级) || \* / % // | 乘、除、取余数、取整除 || + - | 加法、减法 |

- 例如：
  - `2 + 3 * 5 = 17`
  - `(2 + 3) * 5 = 25`
  - `2 * 3 + 5 = 11`
  - `2 * (3 + 5) = 16`

## 算数运算符

计算机，顾名思义就是负责进行 **数学计算** 并且 **存储计算结果** 的电子设备

### 目标

- 算术运算符的基本使用

#### 01. 算数运算符

- 算数运算符是 **运算符** 的一种
- 是完成基本的算术运算使用的符号，用来处理四则运算

| 运算符 | 描述 | 实例 || :--: | :--: | :--: || + | 加 | 10 + 20 = 30 || - | 减 | 10 - 20 = -10 || \* | 乘 | 10 \* 20 = 200 || / | 除 | 10 / 20 = 0.5 || // | 取整除 | 返回除法的整数部分（商） 9 // 2 输出结果 4 || % | 取余数 | 返回除法的余数 9 % 2 = 1 || \*\* | 幂 | 又称次方、乘方，2 \*\* 3 = 8 |

- 在 Python 中 `*` 运算符还可以用于字符串，计算结果就是字符串重复指定次数的结果

```
python In [1]: "-" * 50 Out[1]: '-----'
```

#### 02. 算数运算符的优先级

- 和数学中的运算符的优先级一致，在 Python 中进行数学计算时，同样也是：

- 先乘除后加减**
  - 同级运算符是 **从左至右** 计算
  - 可以使用 `()` 调整计算的优先级
- 以下表格的算数优先级由高到最低顺序排列

| 运算符 | 描述 || -- | :-- || \*\* | 幂 (最高优先级) || \* / % // | 乘、除、取余数、取整除 || + - | 加法、减法 |

- 例如：
  - `2 + 3 * 5 = 17`
  - `(2 + 3) * 5 = 25`
  - `2 * 3 + 5 = 11`
  - `2 * (3 + 5) = 16`

## 变量的基本使用

程序就是用来处理数据的，而变量就是用来存储数据的

### 目标

- 变量定义
- 变量的类型
- 变量的命名

## 01. 变量定义

- 在 Python 中，每个变量 在使用前都必须赋值，变量 赋值以后 该变量 才会被创建
- 等号 (=) 用来给变量赋值
  - = 左边是一个变量名
  - = 右边是存储在变量中的值

```
python 变量名 = 值
```

变量定义之后，后续就可以直接使用了

### 1) 变量演练1 —— iPython

```
```python
```

定义 qq_number 的变量用来保存 qq 号码

```
In [1]: qq_number = "1234567"
```

输出 qq_number 中保存的内容

```
In [2]: qq_number Out[2]: '1234567'
```

定义 qq_password 的变量用来保存 qq 密码

```
In [3]: qq_password = "123"
```

输出 qq_password 中保存的内容

```
In [4]: qq_password Out[4]: '123' ```
```

使用交互式方式，如果要查看变量内容，直接输入变量名即可，不需要使用 `print` 函数

2) 变量演练 2 —— PyCharm

```
```python
```

#### 定义 qq 号码变量

```
qq_number = "1234567"
```

#### 定义 qq 密码变量

```
qq_password = "123"
```

#### 在程序中，如果要输出变量的内容，需要使用 `print` 函数

```
print(qqnumber) print(qqpassword) ```
```

使用解释器执行，如果要输出变量的内容，必须要使用 `print` 函数

### 3) 变量演练 3 —— 超市买苹果

- 可以用 其他变量的计算结果 来定义变量
- 变量定义之后，后续就可以直接使用了

#### 需求

- 苹果的价格是 8.5 元/斤
- 买了 7.5 斤 苹果
- 计算付款金额

```
```python
```

定义苹果价格变量

```
price = 8.5
```

定义购买重量

```
weight = 7.5
```

计算金额

```
money = price * weight
```

```
print(money) ```
```

思考题

- 如果 只要买苹果，就返 5 块钱
- 请重新计算购买金额

```
```python
```

#### 定义苹果价格变量

```
price = 8.5
```

# 定义购买重量

```
weight = 7.5
```

# 计算金额

```
money = price * weight
```

# 只要买苹果就返 5 元

```
money = money - 5 print(money) ``
```

提问

- 上述代码中，一共定义有几个变量？
  - 三个：price / weight / money
- money = money - 5 是在定义新的变量还是在使用变量？
  - 直接使用之前已经定义的变量
  - 变量名 只有在 第一次出现 才是 定义变量
  - 变量名 再次出现，不是定义变量，而是直接使用之前定义过的变量
- 在程序开发中，可以修改之前定义变量中保存的值吗？
  - 可以
  - 变量中存储的值，就是可以 变 的

## 02. 变量的类型

- 在内存中创建一个变量，会包括：
  1. 变量的名称
  2. 变量保存的数据
  3. 变量存储数据的类型
  4. 变量的地址（标示）

### 2.1 变量类型的演练 —— 个人信息

需求

- 定义变量保存小明的个人信息
- 姓名：小明
- 年龄：18 岁
- 性别：是男生
- 身高：1.75 米
- 体重：75.0 公斤

利用 单步调试 确认变量中保存数据的类型

提问

1. 在演练中，一共有几种数据类型？
  - 4 种
  - str —— 字符串
  - bool —— 布尔（真假）
  - int —— 整数
  - float —— 浮点数（小数）
2. 在 Python 中定义变量时需要指定类型吗？
  - 不需要
  - Python 可以根据 = 等号右侧的值，自动推导出变量中存储数据的类型

### 2.2 变量的类型

- 在 Python 中定义变量是 不需要指定类型（在其他很多高级语言中都需要）
- 数据类型可以分为 数字型 和 非数字型
- 数字型
  - 整型 (int)
  - 浮点型 (float)
  - 布尔型 (bool)
    - 真 True 非 0 数 —— 非零即真
    - 假 False 0
  - 复数型 (complex)
    - 主要用于科学计算，例如：平面场问题、波动问题、电感电容等问题
- 非数字型
  - 字符串
  - 列表
  - 元组
  - 字典

提示：在 Python 2.x 中，整数 根据保存数值的长度还分为： \* int（整数） \* long（长整数）

- 使用 type 函数可以查看一个变量的类型

```
python In [1]: type(name)
```

### 2.3 不同类型变量之间的计算

#### 1) 数字型变量 之间可以直接计算

- 在 Python 中，两个数字型变量是可以直接进行 算术运算的
- 如果变量是 bool 型，在计算时
  - True 对应的数字是 1
  - False 对应的数字是 0

演练步骤

1. 定义整数 i = 10
2. 定义浮点数 f = 10.5
3. 定义布尔型 b = True
4. 在 iPython 中，使用上述三个变量相互进行算术运算

2) 字符串变量 之间使用 + 拼接字符串

- 在 Python 中，字符串之间可以使用 + 拼接生成新的字符串

```
python In [1]: first_name = "三"

In [2]: last_name = "张"

In [3]: first_name + last_name Out[3]: '三张'``
```

3) 字符串变量 可以和 整数 使用 \* 重复拼接相同的字符串

```
python In [1]: "-" * 50 Out[1]: '-----'
```

4) 数字型变量 和 字符串 之间 不能进行其他计算

```
python In [1]: first_name = "zhang"

In [2]: x = 10
```

In [3]: x + first\_name

TypeError: unsupported operand type(s) for +: 'int' and 'str' 类型错误: + 不支持的操作类型: int 和 str``

2.4 变量的输入

- 所谓 输入，就是用代码 获取 用户通过 键盘 输入的信息
- 例如：去银行取钱，在 ATM 上输入密码
- 在 Python 中，如果要获取用户在 键盘 上的输入信息，需要使用到 input 函数

1) 关于函数

- 一个 提前准备好的功能(别人或者自己写的代码)，可以直接使用，而 不用关心内部的细节
- 目前已经学习过的函数

| 函数 | 说明 || --- || --- || print(x) | 将 x 输出到控制台 || type(x) | 查看 x 的变量类型 |

2) input 函数实现键盘输入

- 在 Python 中可以使用 input 函数从键盘等待用户的输入
- 用户输入的 任何内容 Python 都认为是一个 字符串
- 语法如下：

```
python 字符串变量 = input("提示信息: ")
```

3) 类型转换函数

| 函数 | 说明 || --- || --- || int(x) | 将 x 转换为一个整数 || float(x) | 将 x 转换到一个浮点数 |

4) 变量输入演练 —— 超市买苹果增强版

需求

- 收银员输入 苹果的价格，单位：元 / 斤
- 收银员输入 用户购买苹果的重量，单位：斤
- 计算并且 输出 付款金额

演练方式 1

```
python
```

1. 输入苹果单价

```
price_str = input("请输入苹果价格: ")
```

2. 要求苹果重量

```
weight_str = input("请输入苹果重量: ")
```

3. 计算金额

1> 将苹果单价转换成小数

```
price = float(price_str)
```

2> 将苹果重量转换成小数

```
weight = float(weight_str)
```

3> 计算付款金额

```
money = price * weight

print(money)``
```

提问

- 演练中，针对 价格 定义了几个变量？
  - 两个
  - price\_str 记录用户输入的价格字符串
  - price 记录转换后的价格数值
- 思考 —— 如果开发中，需要用户通过控制台 输入 很多个 数字，针对每一个数字都要定义两个变量，方便吗？

演练方式 2 —— 买苹果改进版

- 定义 一个 浮点变量 接收用户输入的同时，就使用 float 函数进行转换

```
python price = float(input("请输入价格:"))
```

- 改进后的好处：
- 节约空间，只需要为一个变量分配空间
- 起名字方便，不需要为中间变量起名字
- 改进后的“缺点”：
- 初学者需要知道，两个函数能够嵌套使用，稍微有一些难度

提示

- 如果输入的不是一个数字，程序执行时会出错，有关数据转换的高级话题，后续会讲！

2.5 变量的格式化输出

苹果单价 5.00 元 / 斤，购买了 5.00 斤，需要支付 45.00 元

- 在 Python 中可以使用 print 函数将信息输出到控制台
- 如果希望输出文字信息的同时，一起输出 数据，就需要使用到 格式化操作符
- % 被称为 格式化操作符，专门用于处理字符串中的格式
  - 包含 % 的字符串，被称为 格式化字符串
  - % 和不同的 字符 连用，不同类型的数据 需要使用 不同的格式化字符

| 格式化字符 | 含义 | --- | --- | | %s | 字符串 | | %d | 有符号十进制整数，%06d 表示输出的整数显示位数，不足的地方使用 0 补全 | | %f | 浮点数，%.2f 表示小数点后只显示两位 | | %% | 输出 % |

- 语法格式如下：
- ```
```python print("格式化字符串" % 变量1)

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

格式化输出演练 —— 基本练习

需求

1. 定义字符串变量 name，输出 我的名字叫 小明，请多多关照！
2. 定义整数变量 student\_no，输出 我的学号是 000001
3. 定义小数 price、weight、money，输出 苹果单价 9.00 元 / 斤，购买了 5.00 斤，需要支付 45.00 元
4. 定义一个小数 scale，输出 数据比例是 10.00%

```
python print("我的名字叫 %s，请多多关照！" % name) print("我的学号是 %06d" % student_no) print("苹果单价 %.02f 元 / 斤，购买 %.02f 斤，需要支付 %.02f 元" % (price, weight, money)) print("数据比例是 %.02f%" % (scale * 100))
```

课后练习 —— 个人名片

需求

- 在控制台依次提示用户输入：姓名、公司、职位、电话、邮箱
- 按照以下格式输出：

```
...

公司名称

姓名 (职位)

电话: 电话 邮箱: 邮箱

...
```

实现代码如下：

```
```python """ 在控制台依次提示用户输入：姓名、公司、职位、电话、电子邮箱 """ name = input("请输入姓名：") company = input("请输入公司：") title = input("请输入职位：") phone = input("请输入电话：") email = input("请输入邮箱：")

print(""" 50) print(company) print() print("%s (%s)" % (name, title)) print() print("电话： %s" % phone) print("邮箱： %s" % email) print(""" * 50)

...```
```