

AIDTN2202班级训练营-Day02

1 课程内容回顾

1、python是什么？

Python是一门简洁、免费开源、面向对象、解释型的计算机编程语言【后端】。

2、Python程序的执行方式

1. 创建文件-命令方式执行

文件后缀: .py

Linux命令:

pwd

cd 路径

ls

cat xxx.py

python3 xxx.py

2. python交互环境

1. 终端: python3

2. 进入: >>> python 语句

3. 退出: exit()/Ctrl + Z

3. pycharm 软件

1、创建项目: open-->文件夹

2、创建文件夹: 右击 --> New --> Dir --> 命名

3、创建python文件: 右击 --> New --> Python File --> 命名

4、编写代码

5、运行: 编写代码区 --> 右击 --> run --> 下方的run窗口

3、与计算机交互

- 输入函数: 变量名 = input("提示信息") 【等待用户输入】
- 输出函数: print(数据, ..., sep=' ', end='\n') 【'\n' 换行符】

注释: 以 # 开头, 给人看的

4、变量

- 功能: 关联一个数据
- 命名规则:
 - 1、由数字、字母、下划线组成
 - 2、数字不能开头
 - 3、区分大小写
 - 4、不能使用Python关键字【深蓝色加粗】

- 赋值
 - 变量名 = 数据

5、数据类型

- int - 整型（描述整数）
- float - 浮点数值型（描述小数）
- str - 字符串（描述文本信息）
 - " " ' '

6、数据类型转换

- int(n)
- float(n)

7、运算符

- 算术
 - + - * \ ** \ \ %
- 增强
 - += -= *= \= **= \= %=
- 比较
 - < <= > >= == !=
 - 返回值: True/False
- 逻辑
 - 并且 and
 - 或者 or
 - 非/不是 not
 - 返回值: True/False

8、if 语句

- 功能：有选择性的执行某些语句
- 说明：代码从上向下执行【顺序结构】
- 语法

if 条件1:

 语句1

elif 条件2:

 语句2

...

else:

 语句n

1、if 语句

总结:

- 1、1种情况: if
- 2、2种情况: if + else/if + elif

3、3种或3种以上情况：if + elif + else

4、正确/错误分开、再判断：if + if

2、循环

1、while 语句

- 功能：根据条件【重复】执行某些语句
- 语法：

```
初始条件
while 循环条件: # 循环条件为True
    循环体
    变化条件
else: # 可省略, 循环条件不成立时执行
    语句n
```

- while 死循环
 - 概念：循环条件一直成立。
 - 语法：

```
while True:
    循环体
```

- 说明：
 - 一般会 and break 语句【终止当前的循环】结合使用。

练习：猜数程序

- 随机产生一个数，循环猜测，直到猜对结束，提示大了、小了，猜对打印猜了多少次。
-

2、for 语句

- 功能：用来遍历可迭代对象中的数据
- 可迭代对象：可以依次获取数据的对象
 - str/list/dict、range函数
- 语法：

```
for 变量 in 可迭代对象:
    循环体
else: # 可省略, 可迭代对象中无数据则执行
    语句
```

range函数

- 功能：生成一系列整数的可迭代对象。
- 格式：range([start], stop, [step])
- 参数：

- start: 起始值，默认为0【可省略】
- stop: 终止值，不包含【必须】
- step: 步长/间隔，默认为1【可省略】
 - 正：正向取值（从小到大）
 - 负：反向取值（从大到小）
- 注意：以下情况无数据
 - 1、步长：从小到大取值，start-stop取值是从大到小【方向相反】
 - 2、start与stop值一致

总结

1. while：用于已知/未知循环次数

while True

2. for：用于已知循环次数【推荐】

程序执行结构：

1. 顺序结构：默认的，代码从上向下执行
2. 选择结构：if
3. 循环结构：while/for

3、容器

包括：str/list/dict/tuple/set

1、str - 字符串

- 表示：' ' " " '""' ""' ""' ""'
- 索引

字符串	p	y	t	h	o	n
正向	0	1	2	3	4	5
反向	-6	-5	-4	-3	-2	-1

- 格式：
 - 字符串[索引值]
- 功能：获取字符串中的单个字符
- 切片
 - 功能：获取字符串中的多个字符
 - 语法：
 - str[[start]: [stop] [: step]]
 - 参数
 - start: 起始索引值，默认为最开头【省略】

- stop: 结束索引值，默认为最末尾，不包含【省略】
- step: 步长，默认为1 【省略】
 - 正值：正向切片，从左到右
 - 负值：反向切片，从右到左
- 字符串方法
 - str.strip([char]) 去除字符串指定的字符（默认去空白）
 - str.replace(old, new, count) 替换指定个数的旧字符为新的字符