变量与数据类型

近屿智能 JYI
Just the Yarest Intelligence

目录 CONTENTS

CONTENT

近屿智能 JYI
Just the Yarest Intelligence

01

变量与标识符

变量的定义与本质

01

变量是程序中用于存储数据值的名称，作为内存地址的标签。

```
A2 > test.py > ...
1    a = 1
2    1 = 1
3    壹 = 1
4
```

变量命名规则（标识符）

01

命名必须以字母或下划线开头，可接字母、数字或下划线。

02

避免使用Python关键字，且对大小写敏感。

03

推荐使用有意义的名称，并遵循下划线命名法，如student_name。

02

六大数据类型

整数（int）

整数的定义与特点

整数包括正整数、负整数和零，没有小数部分。

```
# 整数变量
num1 = 5
num2 = 10

# 打印变量
print("整数变量:")
print("num1:", num1)
print("num2:", num2)
```

浮点数（float）

浮点数的定义与应用

浮点数可以表示小数，用于需要小数精确度的计算。

```
# 浮点数变量
float1 = 3.14
float2 = 2.5

# 打印变量
print("浮点数变量:")
print("float1:", float1)
print("float2:", float2)
```

字符串（str）

字符串的定义与操作

字符串用于存储文本数据，
由单引号或双引号括起。

```
# 字符串变量
string1 = "Hello"
string2 = "World"

# 打印变量
print("字符串变量:")
print("string1:", string1)
print("string2:", string2)
```

列表（list）

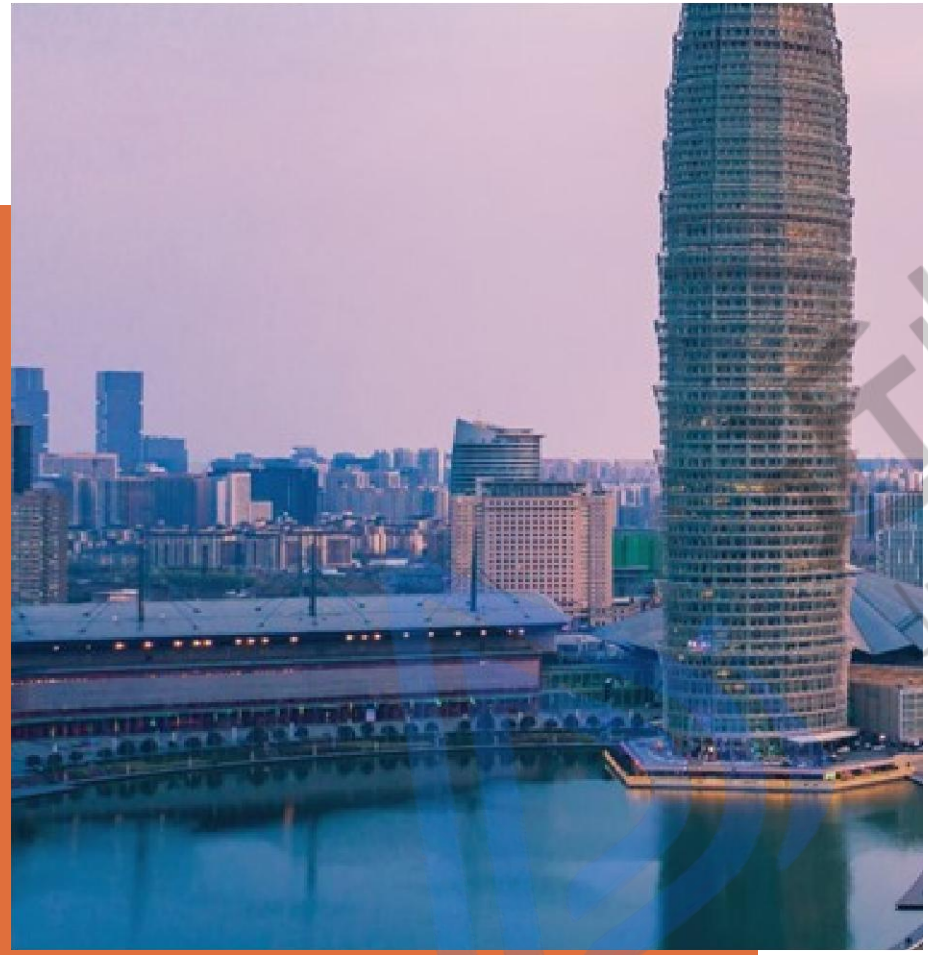
列表是有序的元素集合，可以包含不同类型的元素。

```
# 列表变量
list1 = ["Hello"]
list2 = ["World"]

# 打印变量
print("列表变量:")
print("list1:", list1)
print("list2:", list2)
```

支持元素的添加、删除和修改，是常用的数据结构。

字典（dict）



字典的定义与用途

字典是键值对集合，键唯一，用于存储映射关系。

适用于需要快速查找和更新数据的场景。

```
# 字典变量
dict1 = {"key1": "Hello"}
dict2 = {"key2": "World", "ll": []}

# 打印变量
print("字典变量:")
print("dict1:", dict1)
print("dict2:", dict2)
```

布尔型（bool）

```
# 布尔值变量
bool1 = True
bool2 = False

# 打印变量
print("布尔值变量:")
print("bool1:", bool1)
print("bool2:", bool2)
```

01 布尔型只有True和False两个值，用于逻辑判断。

02 控制程序流程，如条件语句和循环。

03

字符串与格式化输出

字符串基础

字符串的定义

字符串是编程语言中表示文本的数据类型。

```
# 字符串变量
string1 = "Hello"
string2 = "World"
```

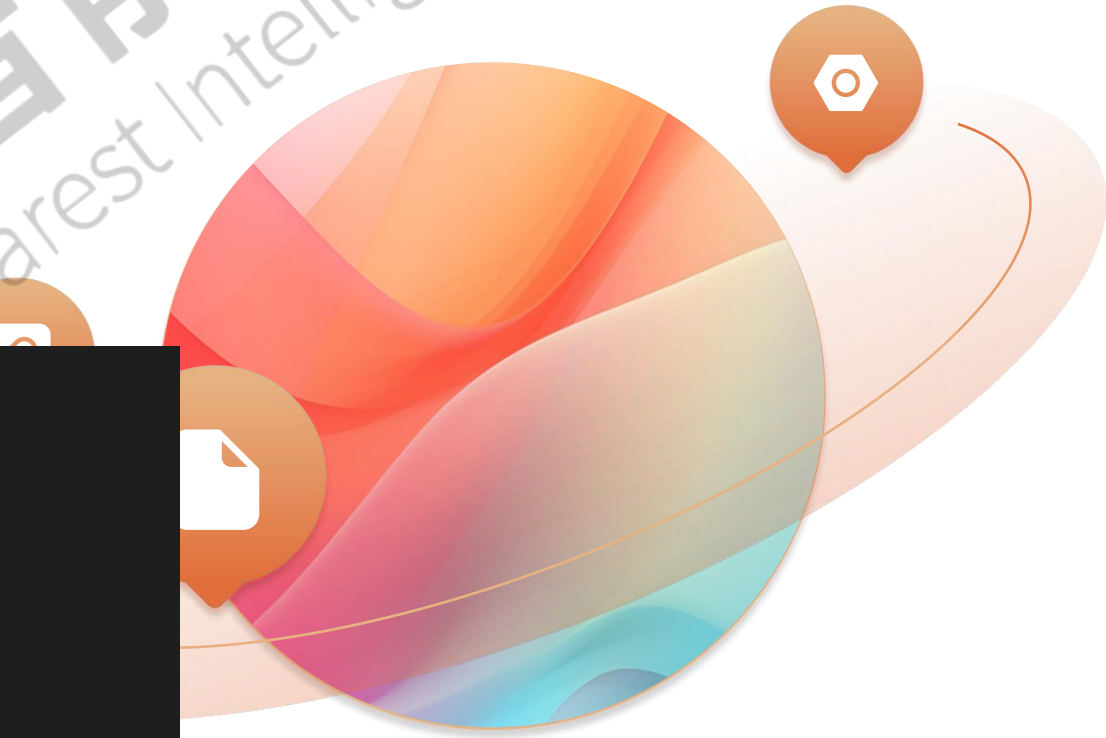
字符串操作

字符串拼接与重复

字符串可以通过+操作符进行拼接，通过*操作符进行重复。

```python
# 字符串拼接
greeting = "Hello"
name = "Alice"
message = greeting + ", " + name + "!"
print(message)  # 输出: Hello, Alice!

# 字符串重复
repeated_message = "Python! " * 3
print(repeated_message)  # 输出: Python! Python! Python!
```



格式化输出

f-string（推荐）

f- string是Python中的字符串格式化方法，简洁且易于阅读。

format()方法

format()方法提供了一种灵活的方式来格式化字符串。

旧式格式化

使用%操作符进行字符串格式化，为旧式方法。

```python
# 字符串格式化
# f-string
name = "Alice"
age = 25
print(f"My name is {name}, and I am {age} years old.")
# 输出: My name is Alice, and I am 25 years old.

# format 方法
name = "Alice"
age = 25
print("My name is {}, and I am {} years old.".format(name, age))
# 输出: My name is Alice, and I am 25 years old.

# % 格式化
name = "Alice"
age = 25
print("My name is %s, and I am %d years old." % (name, age))
# 输出: My name is Alice, and I am 25 years old.
```

常见字符串方法

字符串方法

字符串提供了多种方法，如.lower()、.upper()、.strip()等。

```
# 常见字符串方法

# .lower() 方法将字符串转换为小写
text = "Hello, World!"
lowercase_text = text.lower()
print(lowercase_text)  # 输出：hello, world!

# .upper() 方法将字符串转换为大写
text = "Hello, World!"
uppercase_text = text.upper()
print(uppercase_text)  # 输出：HELLO, WORLD!

# .strip() 方法删除字符串两端的空格
text = "   Hello, World!   "
stripped_text = text.strip()
print(stripped_text)  # 输出：Hello, World!
```

近屿智能 JYI
Just the Yarest Intelligence

谢谢大家