CS50P Notes

YiranY

June 15, 2025

The Python Tutorial: `http://docs.python.org/zh-cn/3.13/tutorial/index.html`

Contents

1 Functions, Variables

1.1 Creating Code with Python

Executing `code hello.py` in the terminal will create a new file (only works in VSCode). To run this program, type `Python3 hello.py` in the terminal and then press the enter key.

1.2 Example of Simple Functions

See documentation for `print()` and `input()`.

Some functions take many arguments. We can use a comma to pass in multiple arguments.

`input()` is a function that takes a prompt as an argument and returns a string. In this example, the value returned by `input("What's your name? ")` is assigned to variable `name` (a string).

```
# This hashtag stands for comment in python

# Ask the user for their name
name = input("What's your name? ")

# Print hello and the inputted name
print("hello,", name)
```

If we typed "Dou", the output would be `hello, Dou`.

1.3 Parameters and Arguments

Parameter is a variable defined in a function declaration (the placeholder).

Argument is the actual value passed to the function when it is called.

`print()` automatically (default) includes the argument `end='\n'`. We can provide an argument for `end` such that a new line won't be created.

```
# This program has the same output as previous
name = input("What's your name? ")

print("hello,", end=" ")
# end is a parameter
# end = " " is a argument

print(name)
```

1.4 Strings

A string, known as `str` in Python, is a sequence of text. See documentation on *str*.

```
# Another way to use strings - Formatting Strings
name = input("What's your name? ")

print(f"hello, {name}")
```

This `f` in `print(f"hello, {name}")` indicates an **f-string**. It's a way to embed variables (in this example, `name`) or expressions directly inside a string by placing them inside curly braces {}.

`strip()` method will remove any whitespace at the beginning and end of the string.

`title()` method will capitalize the first letter of each word in the string.

```
# An efficient way
name = input("What's your name? ").strip().title()

print(f"hello, {name}")
```

1.5 Integers

```
x = input("What's x? ")
y = input("What's y? ")

# without converting into int, x and y will be treated as string
z = int(x) + int(y)

print(z)
```

The use of `int(x)` is called "casting".

```
x = int(input("What's x? "))
y = int(input("What's y? "))

print(x + y)
```

This illustrates that we can run functions on functions, from inner to outer.

1.6 Floats

We could do `round(n)` to round a digit to its nearest integer.

```
# Get the floated input
x = float(input("What's x? "))
y = float(input("What's y? "))

# Create a rounded result
z = round(x + y)
```

```
# Round to the nearesr two decimal points
z = round(x / y, 2)

print(z)
```

1.7 Def

A way to create functions.

```
# Create our own function
def hello(to="world"):
    print("hello,", to)

# Output using our own function
name = input("What's your name? ")
hello(name)

# Output by default
hello()
```

This `hello()` function takes a single parameter: a variable called `to`, with a default value "world". When calling `hello(name)`, the computer passes `name` into the function as `to`.