UC San Diego

CSE6R NOTES

Summarized course notes by Emyl Safin bin Mohd Zaky

Instructor: Moshiri Niema

SP23

Table of Contents

The Basics

1.1 Variables

We pass data to the computer through the use of **variables**. Variables can be assigned a value which will have a **basic data type** and can be plugged into expressions like in math. Basic Data Types:

int - Holds any **whole number**.

```
example_one = 9
```

float - Holds any **decimal**.

```
example_two = 3.845
```

bool - Holds either **True** or **False**.

```
example_three = False
```

str - Holds **one or more characters**, represented with double or single quotes. (Note that it is common to call variables of type str as strings)

```
example_four = "This is a string."
```

Note that in Python we can assign multiple variables in one line as shown in the example below.

```
foo, bar, baz = True, "Gauss", -2.3411
```

There is a fifth special basic data type called **NoneType**.

NoneType - When there is no data to be stored in a variable, assigning the value **None**.

```
example_none_type = None
```

1.2 Strings

1.2.1 Zero-based numbering

Consider the string below:

```
lilith = "Daughter of Hatred"
```

In Python, data is **zero-indexed**. Meaning that the **initial element** of a sequence is assigned the **position(index) 0** instead of the position 1.
Hence in the line above the first element or index 0 is D, the third element or index 2 is u, and so forth.

1.2.2 Indexing

We can verify the fact above through the use of the **indexing operator**, which allows us to access elements at specific indexes. For example:

```
lilith = "Daughter of Hatred"
first = lilith[0] # Indexing Operator
print(first)
```

Output

D

Substrings

We call **sequential characters** in a string a **substring**. The indexing operator can be used to extract substrings. In general we do this by:

```
foobar = ??? # Arbitrary string
substring = foobar[start : end + 1]
```

Here start represents the starting position and end is the ending position of the substring.
Note that the +1 is due to the **end index** being **exclusive** in Python.
We can see this work more specifically in the example below:

```
diablo = "Lord of Terror"
substring = diablo[8 : 14]
print(substring)
```

Output

Terror