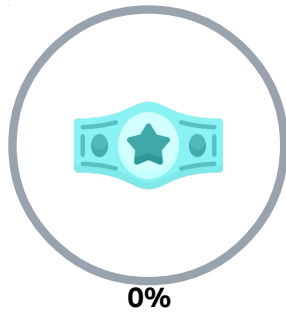


(/)



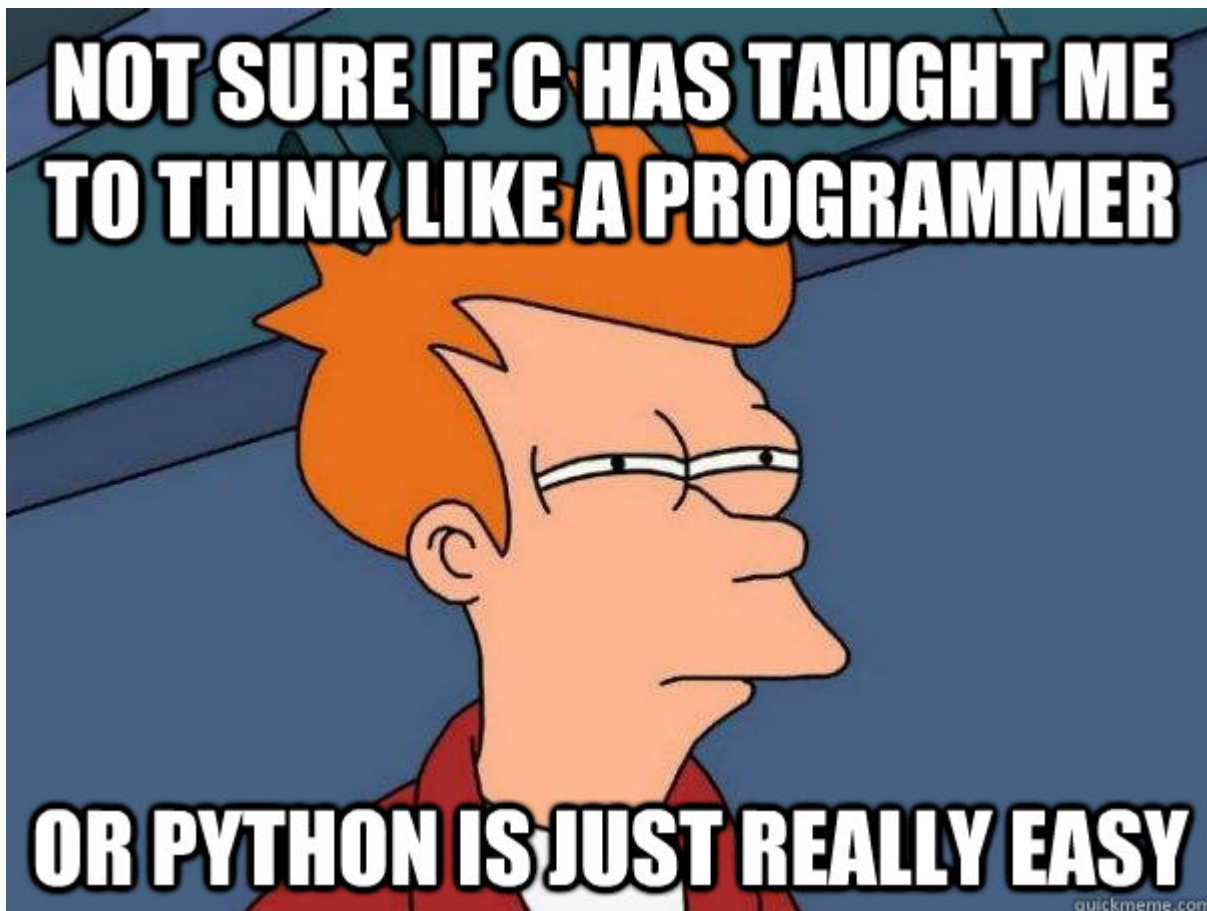
Python - Hello, World

**Python**

↑ Novice

By: Guillaume, CTO at Holberton School

⚙ Weight: 1

☒ Your score will be updated once you launch the project review.

Resources

Read or watch:

- How to install Python (/rltoken/XSnxAE-fxATz8iSacEgEIQ)
- (/). Python for Absolute beginners (/rltoken/4zPH4Tis4yUDPiP1dVoyGg)
- Python Keywords Identifiers (/rltoken/FPyIUx4e33wFkJ4B7DuFA)
- Python Data Types (/rltoken/_CGHpkrUZkbvtnk9qck4sw)
- How to Code in Python 3 (/rltoken/77b1hBlyQsoJnJytVsXibQ)
- The Python tutorial (/rltoken/-bBUU2j37qWUhGgOiJ8TMQ) (*only the first three chapters below*)
 - Whetting Your Appetite (/rltoken/EFfYXskNklw7coW9YDD1eQ)
 - Using the Python Interpreter (/rltoken/rSv7wA-WOZQT11oWzZNO2g)
 - An Informal Introduction to Python (/rltoken/lnzLV68eE-Xb4ibJScpErA) (*Read up until "3.1.2. Strings" included*)
- How To Use String Formatters in Python 3 (/rltoken/1zHzTVKFHVWn5E_VgwL7aA)
- Learn to Program (/rltoken/sXfNvtzduGMjpbbyMYPTw)
- More Control Flow Tools (/rltoken/GnDMSMJ8pKPJhtP62A-v8A) (*Read until "4.6. Defining Functions" included*)
- Myths about Indentation (/rltoken/5dhYoFfcG0bby8n62CDfDg)
- IndentationError - video (/rltoken/iT3YYNBemuyyLubCemDIA)
- How To Use String Formatters in Python 3 (/rltoken/1zHzTVKFHVWn5E_VgwL7aA)
- Learn to Program - video (/rltoken/sXfNvtzduGMjpbbyMYPTw)
- Learn to Program 2 : Looping - video (/rltoken/sXfNvtzduGMjpbbyMYPTw)
- PEP 8 – Style Guide for Python Code (/rltoken/KceEskXAtqD_ESIGOBNfog)

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone

(/rltoken/gmqCtZJozGkxk1twcttLkA), **without the help of Google:**

General

- Why Python programming is awesome
- Who created Python
- Who is Guido van Rossum
- Where does the name 'Python' come from
- What is the Zen of Python
- How to use the Python interpreter
- How to print text and variables using `print`
- How to use strings
- What are indexing and slicing in Python
- What is the official Holberton Python coding style and how to check your code with `PEP 8`
- Why indentation is so important in Python
- How to use the `if`, `if ... else` statements
- How to use comments
- How to affect values to variables
- How to use the `while` and `for` loops
- How to use the `break` and `continue` statements
- How to use `else` clauses on loops
- What does the `pass` statement do, and when to use it
- How to use `range`
- What does `return` a function that does not use any `return` statement
- Scope of variables

- What's a traceback
- (/)
 - What are the arithmetic operators and how to use them

Requirements

Python Scripts

- Recommended editors: `Visual studio code`
- All your files will be interpreted/compiled on Ubuntu 20.04 LTS using `python3` (version 3.4.3)
- All your files should end with a new line
- A `README.md` file at the root of the `alx_python` repo, containing a description of the repository
- A `README.md` file, at the root of the folder of *this* project, is mandatory
- The length of your files will be tested using `wc`

More Info

Zen

The Zen of Python, by Tim Peters

```
Beautiful is better than ugly.  
Explicit is better than implicit.  
Simple is better than complex.  
Complex is better than complicated.  
Flat is better than nested.  
Sparse is better than dense.  
Readability counts.  
Special cases aren't special enough to break the rules.  
Although practicality beats purity.  
Errors should never pass silently.  
Unless explicitly silenced.  
In the face of ambiguity, refuse the temptation to guess.  
There should be one-- and preferably only one --obvious way to do it.  
Although that way may not be obvious at first unless you're Dutch.  
Now is better than never.  
Although never is often better than *right* now.  
If the implementation is hard to explain, it's a bad idea.  
If the implementation is easy to explain, it may be a good idea.  
Namespaces are one honking great idea -- let's do more of those!
```

(/)



Quiz questions

Great! You've completed the quiz successfully! Keep going! ([Show quiz](#)).

Tasks

0. Hello, print

mandatory

Write a Python script that prints exactly "Programming is like building a multilingual puzzle", followed by a new line.

- Use the function `print`

```
guillaume@ubuntu:~/py/$ python3 0-print.py
"Programming is like building a multilingual puzzle
guillaume@ubuntu:~/py/$
```

Repo:

- GitHub repository: `alx_python`
- Directory: `python-hello_world`
- File: `0-print.py`



Check your code

>_ Get a sandbox

0/5 pts

1. Copy - Cut - Paste

mandatory

Complete this 1-edges.py (https://s3.amazonaws.com/alx-intranet.hbtn.io/uploads/text/2021/3/fd5bb0d5f7712e088ad80eec4fe394d036ee7029.py?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIARDDGGGOUSBVO6H7D%2F20230719%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20230719T124103Z&X-Amz-Expires=345600&X-Amz-SignedHeaders=host&X-Amz-Signature=2321a3c7e22ac5161d09c486b67e3543fe8917254a0afa19c10be0f43894b476)

- You can find the source code 1-edges.py (https://s3.amazonaws.com/alx-intranet.hbtn.io/uploads/text/2021/3/fd5bb0d5f7712e088ad80eec4fe394d036ee7029.py?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIARDDGGGOUSBVO6H7D%2F20230719%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20230719T124103Z&X-Amz-Expires=345600&X-Amz-SignedHeaders=host&X-Amz-Signature=2321a3c7e22ac5161d09c486b67e3543fe8917254a0afa19c10be0f43894b476)
- You are not allowed to use any loops or conditional statements
- Your program should be exactly 8 lines long
- `word_first_3` should contain the first 3 letters of the variable `word`
- `word_last_2` should contain the last 2 letters of the variable `word`
- `middle_word` should contain the value of the variable `word` without the first and last letters

```
guillaume@ubuntu:~/py/$ python3 1-edges.py
First 3 letters: Hol
Last 2 letters: on
Middle word: olberto
guillaume@ubuntu:~/py/$ wc -l 1-edges.py
8 1-edges.py
guillaume@ubuntu:~/py/$
```

Repo:

- GitHub repository: `alx_python`
- Directory: `python-hello_world`
- File: `1-edges.py`

Help

Check your code

>_ Get a sandbox

0/8 pts

2. Positive anything is better than negative nothing

mandatory

This program will assign a random signed number to the variable `number` each time it is executed. Complete the source code in order to print whether the number stored in the variable `number` is positive or negative.

- You can find the source code 2-pon.py (https://s3.amazonaws.com/alx-intranet.hbtn.io/uploads/text/2021/3/94656edc7118841481bb3e6396215a78aedd75b2.py?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIARDDGGGOUSBVO6H7D%2F20230719%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20230719T124103Z&X-Amz-Expires=345600&X-Amz-SignedHeaders=host&X-Amz-Signature=e7e6b8c2d7894abc69d803f38ea0f10b1e2cb2c269c775ffe606c8bc3b084e711)
- The variable `number` will store a different value every time you will run this program
- You don't have to understand what `import`, `random`, `randint` do. Please do not touch this code
- The output of the program should be:
 - The number, followed by
 - if the number is greater than 0: is positive
 - if the number is 0: is zero
 - if the number is less than 0: is negative
 - followed by a new line

```
guillaume@ubuntu:~/$ python3 2-positive_or_negative.py
-4 is negative
guillaume@ubuntu:~/$ python3 2-positive_or_negative.py
0 is zero
guillaume@ubuntu:~/$ python3 2-positive_or_negative.py
-3 is negative
guillaume@ubuntu:~/$ python3 2-positive_or_negative.py
-10 is negative
guillaume@ubuntu:~/$ python3 2-positive_or_negative.py
10 is positive
guillaume@ubuntu:~/$ python3 2-positive_or_negative.py
-5 is negative
guillaume@ubuntu:~/$ python3 2-positive_or_negative.py
6 is positive
guillaume@ubuntu:~/$ python3 2-positive_or_negative.py
7 is positive
guillaume@ubuntu:~/$ python3 2-positive_or_negative.py
5 is positive
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx_python`
- Directory: `python-hello_world`
- File: `2-positive_or_negative.py`

[Help](#)[Check your code](#)[>_ Get a sandbox](#)**0/14 pts**

3. The last digit

mandatory

This program will assign a random signed number to the variable `number` each time it is executed. Complete the source code in order to print the last digit of the number stored in the variable `number`.

- You can find the source code 3-last_digit.py (https://s3.amazonaws.com/alx-intranet.hbtn.io/uploads/text/2021/3/b53c4f6618802f61b84b941a758073c8f6426935.py?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIARDDGGGOUSBVO6H7D%2F20230719%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20230719T124103Z&X-Amz-Expires=345600&X-Amz-SignedHeaders=host&X-Amz-Signature=a227cba1efcf9d5da4a908d304e91ea76af0b2ad7d23d4f6e02b807b52491918)
- The variable `number` will store a different value every time you will run this program
- You don't have to understand what `import`, `random.randint` do. **Please do not touch this code.** This line should not change: `number = random.randint(-10000, 10000)`
- The output of the program should be:
 - The string `Last digit of`, followed by
 - the number, followed by
 - the string `is`, followed by the last digit of `number`, followed by
 - if the last digit is greater than 5: the string `and is greater than 5`
 - if the last digit is 0: the string `and is 0`
 - if the last digit is less than 6 and not 0: the string `and is less than 6 and not 0`
 - followed by a new line

```
guillaume@ubuntu:~/$ python3 3-last_digit.py
Last digit of 4205 is 5 and is less than 6 and not 0
guillaume@ubuntu:~/$ python3 3-last_digit.py
Last digit of -626 is -6 and is less than 6 and not 0
guillaume@ubuntu:~/$ python3 3-last_digit.py
Last digit of 1144 is 4 and is less than 6 and not 0
guillaume@ubuntu:~/$ python3 3-last_digit.py
Last digit of -9200 is 0 and is 0
guillaume@ubuntu:~/$ python3 3-last_digit.py
Last digit of 5247 is 7 and is greater than 5
guillaume@ubuntu:~/$ python3 3-last_digit.py
Last digit of -9318 is -8 and is less than 6 and not 0
guillaume@ubuntu:~/$ python3 3-last_digit.py
Last digit of 3369 is 9 and is greater than 5
guillaume@ubuntu:~/$ python3 3-last_digit.py
Last digit of -5224 is -4 and is less than 6 and not 0
guillaume@ubuntu:~/$ python3 3-last_digit.py
Last digit of -4485 is -5 and is less than 6 and not 0
guillaume@ubuntu:~/$ python3 3-last_digit.py
Last digit of 3850 is 0 and is 0
guillaume@ubuntu:~/$ python3 3-last_digit.py
Last digit of 5169 is 9 and is greater than 5
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx_python`
- Directory: `python-hello_world`
- File: `3-last_digit.py`

Help

Check your code

>_ Get a sandbox

0/14 pts

4. Hexadecimal printing

mandatory

Write a program that prints all numbers from 0 to 98 in decimal and in hexadecimal (as in the following example)

- You can only use one `print` function with string format
- You can only use one loop in your code
- You are not allowed to store numbers or strings in a variable
- You are not allowed to import any module

```
guillaume@ubuntu:~/ $ python3 4-print_hexa.py
0 = 0x0
1 = 0x1
2 = 0x2
3 = 0x3
4 = 0x4
5 = 0x5
6 = 0x6
7 = 0x7
8 = 0x8
9 = 0x9
10 = 0xa
11 = 0xb
12 = 0xc
13 = 0xd
14 = 0xe
15 = 0xf
16 = 0x10
17 = 0x11
18 = 0x12
...
96 = 0x60
97 = 0x61
98 = 0x62
guillaume@ubuntu:~/ $
```

Repo:

- GitHub repository: `alx_python`
- Directory: `python-hello_world`
- File: `4-print_hexa.py`

Help

Check your code

>_ Get a sandbox

0/5 pts

5/100...99

mandatory

Write a program that prints numbers from 0 to 99 .

- Numbers must be separated by , , followed by a space
- Numbers should be printed in ascending order, with two digits
- The last number should be followed by a new line
- You can only use no more than 2 print functions with string format
- You can only use one loop in your code
- You are not allowed to store numbers or strings in a variable
- You are not allowed to import any module

```
guillaume@ubuntu:~/$ python3 5-print_comb2.py
00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45,
46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68,
69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91,
92, 93, 94, 95, 96, 97, 98, 99
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: alx_python
- Directory: python-hello_world
- File: 5-print_comb2.py

Help

Check your code

>_ Get a sandbox

0/5 pts

6. Inventing is a combination of brains and materials. The more brains you use, the less material you need

mandatory

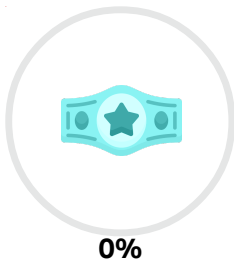
Write a program that prints all possible different combinations of two digits.

- Numbers must be separated by , , followed by a space
- The two digits must be different
- 01 and 10 are considered the same combination of the two digits 0 and 1
- Print only the smallest combination of two digits
- Numbers should be printed in ascending order, with two digits
- The last number should be followed by a new line
- You can only use no more than 3 print functions with string format
- You can only use no more than 2 loops in your code
- You are not allowed to store numbers or strings in a variable
- You are not allowed to import any module

```
guillaume@ubuntu:~/ $ python3 6-print_comb3.py
01, 02, 03, 04, 05, 06, 07, 08, 09, 12, 13, 14, 15, 16, 17, 18, 19, 23, 24, 25, 26, 27, 28,
29, 34, 35, 36, 37, 38, 39, 45, 46, 47, 48, 49, 56, 57, 58, 59, 67, 68, 69, 78, 79, 89
guillaume@ubuntu:~/ $
```

Repo:

- GitHub repository: alx_python
- Directory: python-hello_world
- File: 6-print_comb3.py

[Help](#)[Check your code](#)[>_ Get a sandbox](#)**0/5 pts****Score**

Your score will be updated once you launch the project review.

Please review **all the tasks** before you start the peer review.

[🚀 Review all the tasks](#)[▶▶ Skip this project](#)[Previous project \(/projects/2063\)](/projects/2063)