

Google

🔍 How to print "Hello, World!" from scratch



Google Search

I'm Feeling Lucky

In 33 easy steps

- Patrick Armstrong
Programmer extraordinaire

Why should you listen to me?

1. Described by Neco as “A proper hacker type of guy”
2. Accused of having too clacky a keyboard
3. Use Arch Linux (btw)
4. Lead maintainer of Pippin cosmological pipeline
5. Built many other supernova and cosmological tools:
 - a. SALT2Jacobian
 - b. ShockCooling
 - c. IABCosmo <- PhD project



The ubiquitous “Hello, World!”

Exercise

Use the “print” function to print the line “Hello, World!”.

```
script.py | Python Shell
1 print("Goodbye, World!") | In [1]: |
Solution Run
```

Creating a new Python project

First, create a new folder called `helloworld`.

Second, launch the VS code and open the `helloworld` folder.

Third, create a new `app.py` file and enter the following code and save the file:

```
print('Hello, World!')
```

The `print()` is a built-in function that displays a message on the screen. In this example, it'll show the message `'Hello, World!'`.

C Example

A quick look at the example of `Hello, World!` in C programming, a detailed description is given on the [C Program Structure](#) page.

```
/* Author: www.w3schools.in
Date: 2018-04-28
Description:
Writes the words "Hello World" on the screen */
#include<stdio.h>

int main()
{
    printf("Hello, World!\n");
    getch(); //Use to get one character input from user, and
            //it will not be printed on screen.
    return 0;
}
```

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package helloworldapp;

/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World" to standard output.
 */
public class HelloWorldApp {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Display the string.
    }
}
```

Learning by Examples

With our “Try it Yourself” editor, you can edit Python code and view the result.

Example

```
print("Hello, World!")
```

Try it Yourself

Examples in Each Chapter

Our “Try it Yourself” tool makes it easy to learn C. You can edit code and view the result in your browser.


Example

```
#include <stdio.h>

int main() {
    printf("Hello World!");
    return 0;
}
```

Try it Yourself

Python Online Compiler/Interpreter

We have provided [Python Online Compiler/Interpreter](#) which helps you to **Edit** and **Execute** the code directly from your browser. Try to click the icon  to run the following Python code to print conventional “Hello, World!”.

Below code box allows you to change the value of the code. Try to change the value inside `print()` and run it again to verify the result.

```
# This is my first Python program.
# This will print 'Hello, World!' as the output

print ("Hello, World!");
```

Edit & Run

```
#include <iostream>
using namespace std;

// main() is where program execution begins.
int main() {
    cout << "Hello World"; // prints Hello World
    return 0;
}
```

Live Demo

Hello World using C Programming.

Just to give you a little excitement about **C programming**, I’m going to give you a small conventional C Programming Hello World program. You can try it using [Demo Link](#).

```
#include <stdio.h>

int main() {
    /* my first program in C */
    printf("Hello, World! \n");

    return 0;
}
```

Live Demo

Simple.java

```
class Simple{
    public static void main(String args[]){
        System.out.println("Hello Java");
    }
}
```

Test It Now

Examples in Each Chapter

Our “Try it Yourself” editor makes it easy to learn C++. You can edit C++ code and view the result in your browser.

Example

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello World!";
    return 0;
}
```

Try it Yourself

How to print “Hello, World!” from scratch in 4 easy steps

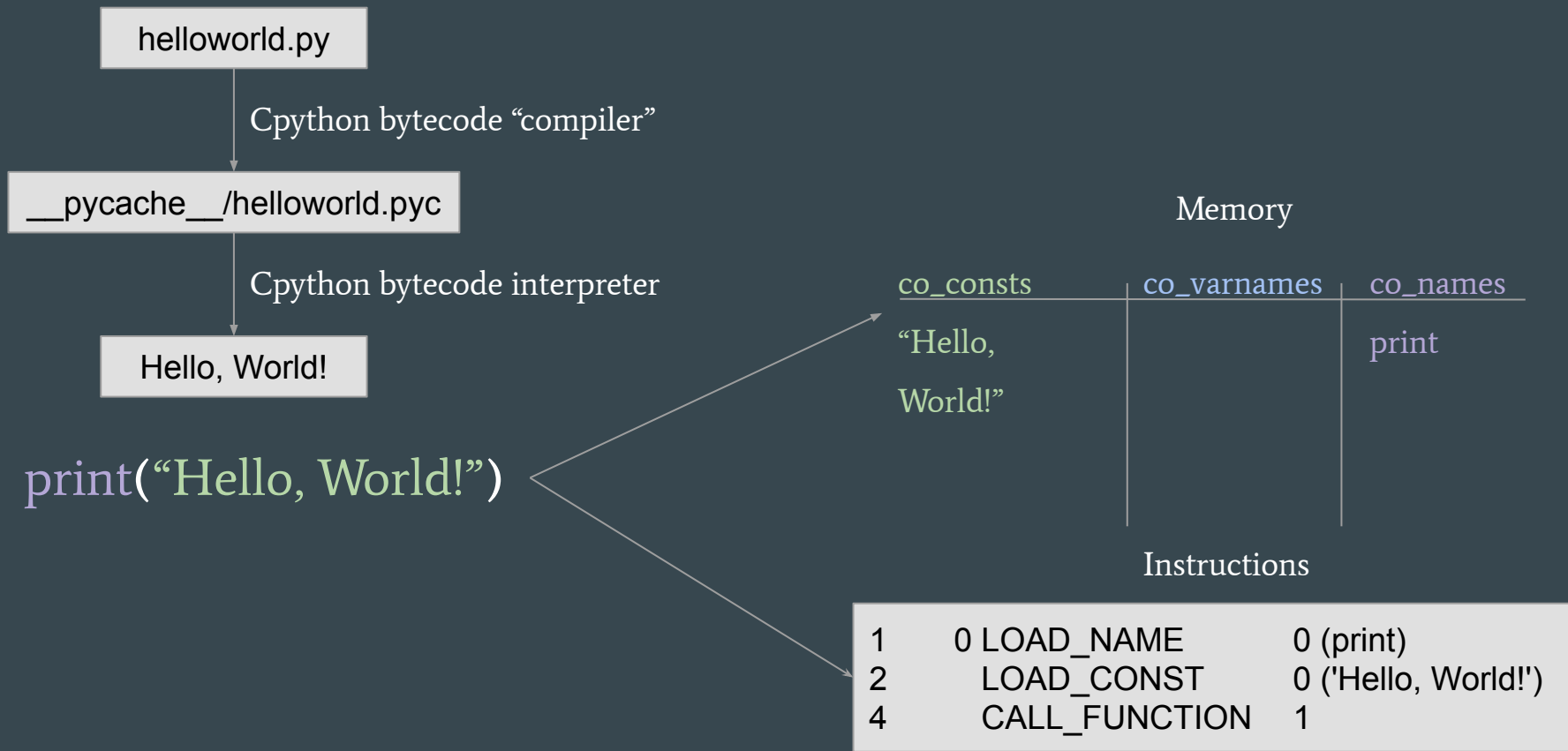
1. Open a new file called `helloworld.py` in your text editor of choice
2. Type `print('Hello, World!')`
3. Open your terminal, navigate to your file, and run `python helloworld.py`
4. You should see `Hello, World!` printed in your terminal!

How to really print “Hello, World!” from scratch in 4 easy steps

1. Open your terminal
2. Run `echo "print('Hello, World!') > helloworld.py"` to create your python script
3. Run `python helloworld.py`
4. You should see `Hello, World!` printed in your terminal!

How does python print “Hello, World!”?

How does python print “Hello, World!”?



How to really, *really* print “Hello, World!” from scratch in 5 easy steps

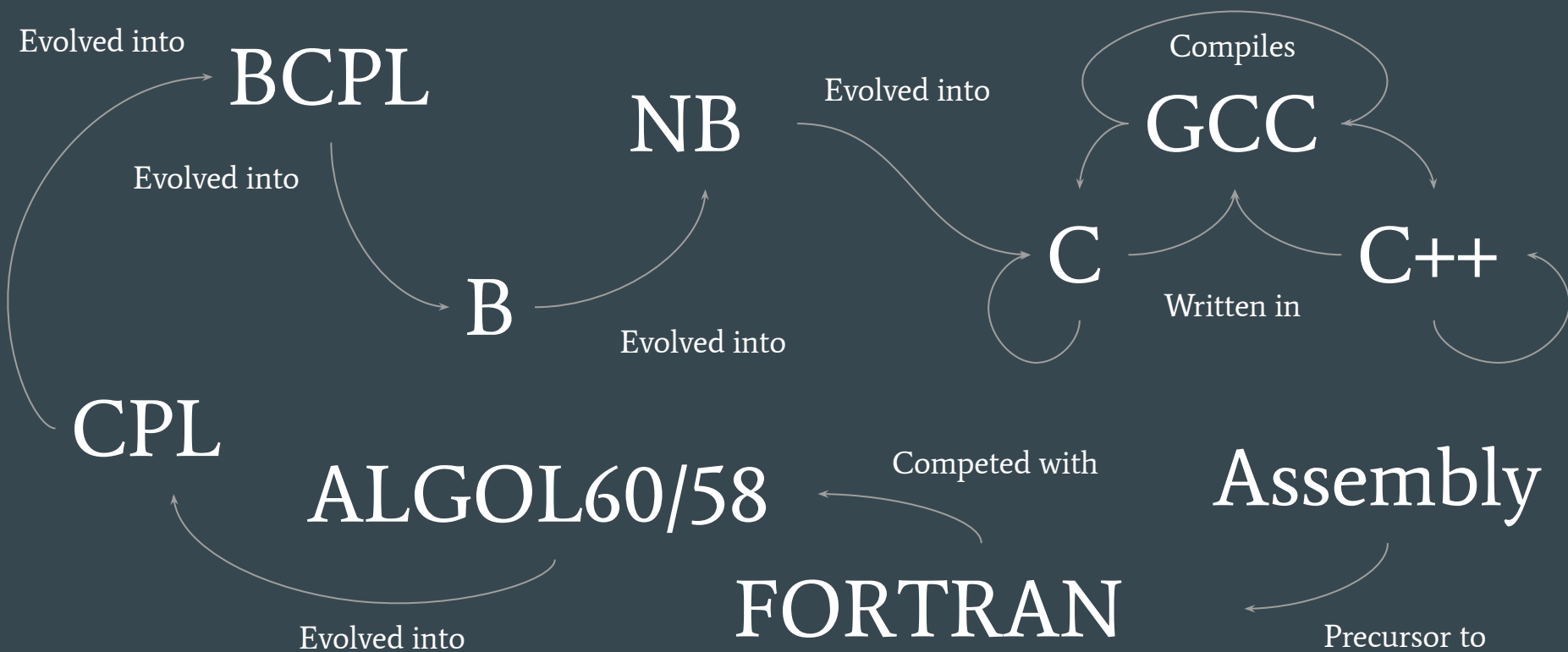
1. Open your terminal
2. Run `echo "print('Hello, World!') > helloworld.py"` to create your python script
3. Compile your script to bytecode via `python -m py_compile helloworld.py`
4. Interpret and run your compiled code via `python __pycache__/helloworld.pyc`
5. You should see `Hello, World!` printed in your terminal!

How do you get python in the first place?

How to really, *really, really* print “Hello, World!” from scratch in 5 easy steps

1. Open your terminal
2. Run `git clone https://github.com/python/cpython``
3. Run `cd cpython``
4. Run `./configure --with-pydebug`` to prepare for compilation
5. Run `make -j8`` to compile python using GCC. You should now have a python executable in your directory
6. Run `echo "print('Hello, World!') > helloworld.py`` to create your python script
7. Compile your script to bytecode via `./python -m py_compile helloworld.py``
8. Interpret and run your compiled code via `./python __pycache__/helloworld.pyc``
9. You should see `Hello, World!` printed in your terminal!

What about C, C++, and GCC?



How to really, *really, really, really* print “Hello, World!” from scratch in 19 easy steps

1. Use Assembly to create FORTRAN which competed with
2. ALGOL58 which evolved into
3. ALGOL60 which evolved into
4. CPL which evolved into
5. BCPL which evolved into
6. B which evolved into
7. NB which evolved into
8. C which was used to create
9. GCC
10. Create a terminal program in C
11. Open your terminal
12. Run `git clone https://github.com/python/cpython``
13. Run `cd cpython``
14. Run `./configure --with-pydebug``
15. Run `make -j8`. You should now have a python script in your directory
16. Run `echo "print('Hello, World!')"` > helloworld.py` to create your python script
17. Compile your script to bytecode via `./python -m py_compile helloworld.py``
18. Interpret and run your compiled code via `./python __pycache__/helloworld.pyc``
19. You should see `Hello, World!` printed in your terminal!

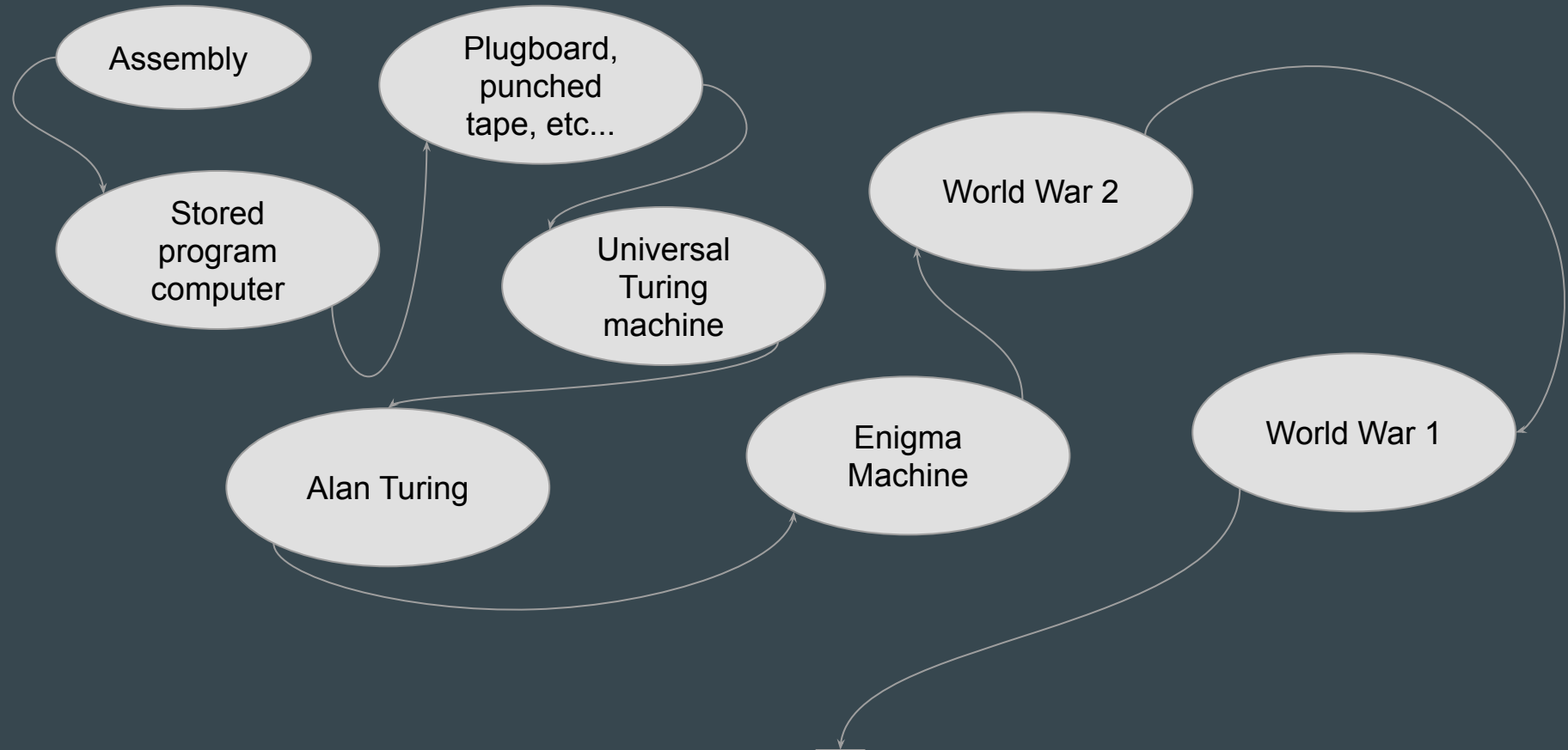
What about `git`, `cd`, `make`, `echo`, etc...

1. `git`, `cd`, `make`, `echo`, etc...: bash commands
2. Bash (Bourne-Again shell) is a unix shell
 - a. Bash replaced Bourne as a free, open-source shell
 - b. Bourne replaced Thompson, the first unix shell
3. Unix shells are command- line interpreters for the Unix operating system
4. Unix is currently written in C, but was originally written in Assembly

How to really, *really, really, really, REALLY* print “Hello, World!” from scratch in 24 easy steps

1. Use Assembly to create UNIX
2. Use Assembly to create the Thompson Shell which evolved into
3. The Bourne Shell which evolved into
4. Bash which allows you to run
5. Bash commands like `git`, `cd`, `make`, and `echo`
6. Use Assembly to create FORTRAN which competed with
7. ALGOL58 which evolved into
8. ALGOL60 which evolved into
9. CPL which evolved into
10. BCPL which evolved into
11. B which evolved into
12. NB which evolved into
13. C which was used to create
14. GCC
15. Create a terminal program in C
16. Open your terminal
17. Run `git clone <https://github.com/python/cpython>`
18. Run `cd cpython`
19. Run `./configure --with-pydebug`
20. Run `make -j8`. You should now have a python script in your directory
21. Run `echo "print('Hello, World!') > helloworld.py` to create your python script
22. Compile your script to bytecode via `./python -m py_compile helloworld.py`
23. Interpret and run your compiled code via `./python __pycache__/helloworld.pyc`
24. You should see `Hello, World!` printed in your terminal!

How is your computer doing any of this?



How to really, *really*, *really*, *really*, *REALLY*, *R E A L L Y* print “Hello, World!” from scratch in 33 easy steps

1. Assassinate an Archduke
2. Survive WW1
3. Survive WW2
4. Create the Enigma Machine
5. Crack the Enigma Machine
6. Develop the Turing Machine
7. Build the first computers
8. Develop electronic computers
9. Create Assembly language
10. Use Assembly to create UNIX
11. Use Assembly to create the Thompson Shell which evolved into
12. The Bourne Shell which evolved into
13. Bash which allows you to run
14. Bash commands like `git`, `cd`, `make`, and `echo`
15. Use Assembly to create FORTRAN which competed with
16. ALGOL58 which evolved into
17. ALGOL60 which evolved into
18. CPL which evolved into
19. BCPL which evolved into
20. B which evolved into
21. NB which evolved into
22. C which was used to create
23. GCC
24. Create a terminal program in C
25. Open your terminal
26. Run `git clone [https://github.com/python/cpython`](https://github.com/python/cpython)
27. Run `cd cpython`
28. Run `./configure --with-pydebug`
29. Run `make -j8`. You should now have a python script in your directory
30. Run `echo "print('Hello, World!')"` > helloworld.py` to create your python script
31. Compile your script to bytecode via `./python -m py_compile helloworld.py`
32. Interpret and run your compiled code via `./python __pycache__/helloworld.pyc`
33. You should see `Hello, World!` printed in your terminal!

Questions?

Compilation

```
python -m py_compile helloworld.py
```

```
__pycache__/helloworld.pyc
```

```
print("Hello, World!")
```

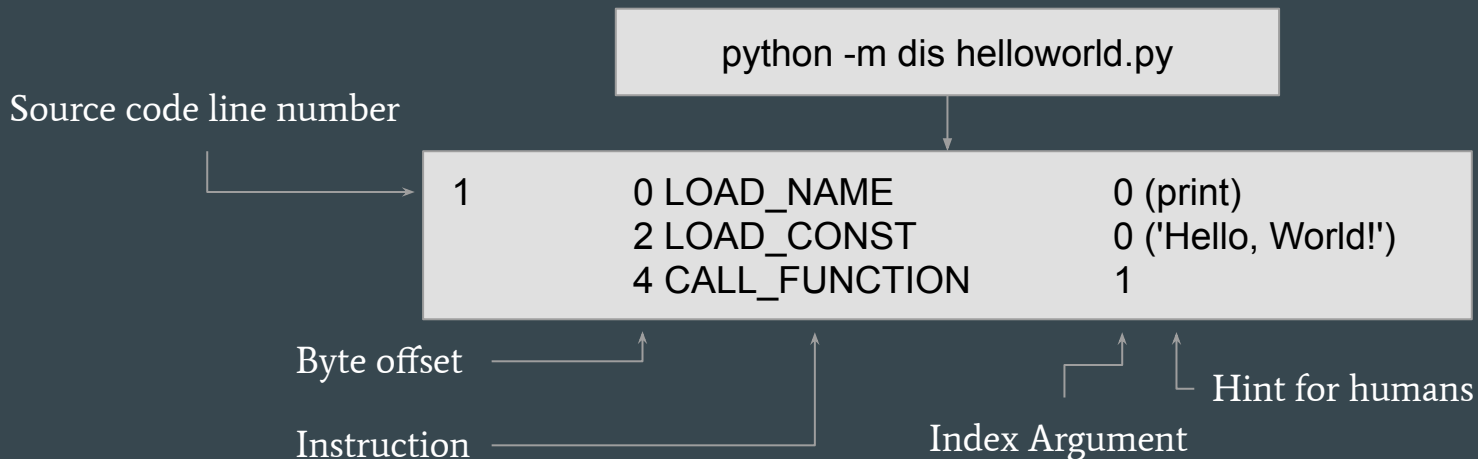
Memory

<u>co_consts</u>	<u>co_varnames</u>	<u>co_names</u>
"Hello, World!"		print

Instructions

1	0	LOAD_NAME	0 (print)
2		LOAD_CONST	0 ('Hello, World!')
4		CALL_FUNCTION	1

Bytecode instructions



Running helloworld.pyc

