



**pythondrops.com**

Curso Básico de Python

*(c) 2019 Cleuton Sampaio*

# **Lição 1: Esquentando o motor**



Este trabalho e todos os seus exemplos, mesmo que não explicitamente citado, estão distribuídos de acordo com a licença Creative Commons Atribuição-CompartilhaIgual 4.0 Internacional. Este é o link para os termos desta licença:

[https://creativecommons.org/licenses/by-sa/4.0/deed.pt\\_BR](https://creativecommons.org/licenses/by-sa/4.0/deed.pt_BR)

Você tem o direito de:

- **Compartilhar** — copiar e redistribuir o material em qualquer suporte ou formato
- **Adaptar** — remixar, transformar, e criar a partir do material para qualquer fim, mesmo que comercial.

Esta licença é aceitável para Trabalhos Culturais Livres.

O licenciante não pode revogar estes direitos desde que você respeite os termos da licença.

De acordo com os termos seguintes:

- **Atribuição** — Você deve dar o crédito apropriado, prover um link para a licença e indicar se mudanças foram feitas. Você deve fazê-lo em qualquer circunstância razoável, mas de nenhuma maneira que sugira que o licenciante apoia você ou o seu uso.
- **CompartilhaIgual** — Se você remixar, transformar, ou criar a partir do material, tem de distribuir as suas contribuições sob a mesma licença que o original.
- **Sem restrições adicionais** — Você não pode aplicar termos jurídicos ou medidas de caráter tecnológico que restrinjam legalmente outros de fazerem algo que a licença permita.

Avisos:

- Você não tem de cumprir com os termos da licença relativamente a elementos do material que estejam no domínio público ou cuja utilização seja permitida por uma exceção ou limitação que seja aplicável.
- Não são dadas quaisquer garantias. A licença pode não lhe dar todas as autorizações necessárias para o uso pretendido. Por exemplo, outros direitos, tais como direitos de imagem, de privacidade ou direitos morais, podem limitar o uso do material.

## Sumário

Introdução.....	4
Histórico.....	5
Popularidade.....	6
Pontos negativos.....	6
O “jeitão” do Python.....	7
Plataforma para rodar Python.....	10
PythonAnywhere.....	11
Codenvy.....	15
No seu computador.....	19
Exercício.....	20

## Introdução

Bom, você vai aprender a programar em Python, certo? Antes de mais nada, vamos deixar claras algumas considerações:

1. Este é um curso básico, que lhe ensinará os fundamentos da linguagem Python e como implementar algoritmos utilizando-a;
2. Assumo que você já conhece alguma linguagem de programação;
3. Nenhuma garantia é oferecida de que você realmente aprenderá o que estou ensinando. Depende apenas do seu próprio esforço.

Outra coisa que quero dizer é que eu não sou um “Pythonista”

(<http://binhminhcs.blogspot.com/2011/12/python-pythonic-pythoneer-pythonist.html>), ou seja, não sou um aficionado por Python, e a considero apenas como uma ferramenta para atingir meus objetivos. Portanto, nem sempre minhas soluções são totalmente “Pythônicas” (<https://blog.startifact.com/posts/older/what-is-pythonic.html>), embora eu procure seguir o guia de estilo e os costumes dos programadores Python (<https://www.python.org/dev/peps/pep-0008/>).

Não sou membro de comunidades Python, alías, de nenhuma comunidade das linguagens que eu sei (C, C++, R, Java, Javascript etc).

Embora não condene quem seja “Pythonista”, eu tenho algumas reservas quanto ao comportamento dessas pessoas.

## Histórico

É preciso saber o histórico da linguagem Python, que atualmente está na **versão 3** (3.7.2 em Janeiro de 2019). Para saber a versão mais atual da Linguagem:

<https://www.python.org/downloads/release>

A versão 2.x do Python ainda existe e tem atualizações de segurança e bug fixes. A última versão é:

<https://www.python.org/downloads/release/python-2715/>

### Devo usar Python 2 ou 3?

Boa pergunta! Você deve aprender Python 3, que é a versão atual da linguagem, e utilizá-la em seus projetos. Python 2 só faz sentido se for um projeto legado. Depois de aprender Python 3, é interessante ver as diferenças entre as duas versões, pois sempre pode ser interessante.

<https://wiki.python.org/moin/Python2orPython3>

A Wikipedia tem um excelente artigo histórico sobre Python:

*“Python é uma linguagem de programação de alto nível, interpretada, de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por **Guido van Rossum** em 1991. Atualmente possui um modelo de desenvolvimento comunitário, aberto e gerenciado pela organização sem fins lucrativos Python Software Foundation. Apesar de várias partes da linguagem possuírem padrões e especificações formais, a linguagem como um todo não é formalmente especificada. O padrão de facto é a implementação **CPython**.*

*A linguagem foi projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional. Prioriza a legibilidade do código sobre a velocidade ou expressividade. Combina uma sintaxe concisa e clara com os recursos poderosos de sua biblioteca padrão e por módulos e frameworks desenvolvidos por terceiros.*

*Python é uma linguagem de propósito geral de alto nível, multiparadigma, suporta o paradigma orientado a objetos, imperativo, funcional e procedural. Possui tipagem dinâmica e uma de suas principais características é permitir a fácil leitura do código e exigir poucas linhas de código se comparado ao mesmo programa em outras linguagens. Devido às suas características, ela é*

*principalmente utilizada para processamento de textos, dados científicos e criação de CGIs para páginas dinâmicas para a web. Foi considerada pelo público a 3ª linguagem "mais amada", de acordo com uma pesquisa conduzida pelo site Stack Overflow em 2018, e está entre as 5 linguagens mais populares, de acordo com uma pesquisa conduzida pela RedMonk.”*

## Popularidade

De acordo com a lista TIOBE (<https://www.tiobe.com/tiobe-index/>) de Dezembro de 2018, Python ultrapassou C++ e se tornou a terceira linguagem de programação mais popular do mundo.

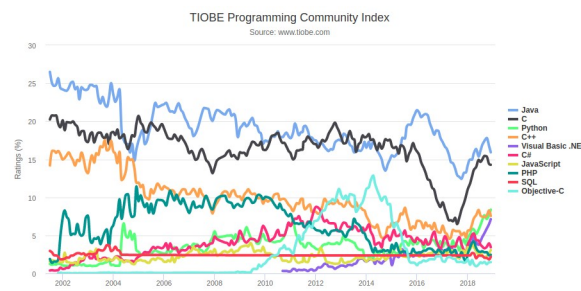


Illustration 1: (c) TIOBE

## Pontos negativos

Entenda que este tópico é apenas minha opinião, portanto, você deve formar a sua. Mas não posso deixar de expressar o que considero como pontos negativos na linguagem Python.

1. **Upgrade destrutivo:** Do Python 2 para o Python 3 muita coisa mudou, causando quebra de compatibilidade;
2. **Identação forçada:** Não adianta reclamar, pois em Python a indentação é sintática! Embora isso seja um pouco desagradável e, por que não dizer, inseguro, deixa o código mais limpo, sem aquele monte de chaves (“{ }”), muito populares em outras linguagens;
3. **Comunidade dogmática:** A comunidade Python é um pouco “dogmática” e segue preceitos quase religiosos, tendendo a desprezar quem não os segue. Isso não é característica única da linguagem Python, pois vejo o mesmo comportamento em outras linguagens, em especial: Lua e Java;
4. **Desempenho poderia melhorar:** Há muito questionamento e blá-blá-blá sobre esta questão, mas alguns sites de benchmark mostram que certas soluções em Python podem ter um

desempenho pior do que em Java (e C++):

<https://benchmarksgame-team.pages.debian.net/benchmarksgame/faster/python.html>

Nenhum destes pontos me impede de afirmar que hoje Python é a minha ferramenta favorita, tanto para aplicações de datascience (Bigdata, Machine Learning etc) como para aplicações IoT e Web.

## O “jeitão” do Python

Ok, bora mostrar alguma coisa em Python? Para começar, vamos fazer o mesmo código em Java. Vamos supor que um Professor goste de arredondar as notas dos alunos da seguinte forma:

- Se a nota for menor ou igual a 60, nenhum arredondamento é feito;
- Se a diferença entre a nota e o próximo múltiplo de 5 superior a ela for menor que 3, então arredonde para o próximo múltiplo de 5;

Então, vejamos um exemplo de notas originais e arredondadas:

Nota original	Nota final
62	62
57	57
83	<b>85</b>
91	91
74	<b>75</b>
63	<b>65</b>

As notas são números inteiros, passados por argumento de linha de comando, para simplificar.

Em Java tradicional:

```
import java.util.Arrays;
public class Grade {
    public static void main(String [] args) {
        int [] notas = new int[args.length];
        for (int x=0; x<notas.length; x++) {
            int nota = Integer.parseInt(args[x]);
            if (nota > 60) {
                int multiplo = ((nota/5)+1)*5;
                if ((multiplo - nota)<3) {
                    notas[x] = multiplo;
                }
                else {
                    notas[x] = nota;
                }
            }
            else {
                notas[x] = nota;
            }
        }
        System.out.println(Arrays.toString(notas));
    }
}
```

Eis a compilação e execução deste código:

```
javac Grade.java
java Grade 62 57 83 91 74 63
[62, 57, 85, 91, 75, 65]
```

Ok. Agora, vamos ver como eu faria isso em Python:

```
import sys
notas = list(map(int,sys.argv[1:]))
novas = [(lambda ng: ((int(ng/5)+1)*5) if ng > 60 and ((int(ng/5)+1)*5) \
- ng < 3 else ng)(ng) for ng in notas]
print(novas)
```

Agora, vejamos a compilação e execução deste script:

```
python grade.py 62 57 83 91 74 63
[62, 57, 85, 91, 75, 65]
```

Hmmmmm. Gostou? Estranhou? O que achou?



A primeira coisa que quero te mostrar é o caractere de continuação contra-barra (“\”) no final da terceira linha. Em Python, podemos controlar a quebra de linha inserindo uma contra-barra onde queremos separar.

Olhando assim, parece que o código Java é mais simples de entender do que o código Python, que parece um pouco confuso, certo? Mas essa é a maneira “pythonica” de escrever código, utilizando expressão **lambda**, um recurso muito utilizado no estilo de **programação funcional**.

*Wikipedia: Em ciência da computação, programação funcional é um paradigma de programação que trata a computação como uma avaliação de funções matemáticas e que evita estados ou dados mutáveis. Ela enfatiza a aplicação de funções, em contraste da programação imperativa, que enfatiza mudanças no estado do programa.*

Resolvemos o problema utilizando apenas funções, sem “loops” explícitos, portanto, simplificando-o.

### O que vimos até agora?

1. Python é uma linguagem de programação interpretada, ou pseudo-compilada em uma só passagem;
2. Os tipos de dados das variáveis são inferidos a partir dos valores atribuídos;
3. Em Python usamos muitos recursos de programação funcional, em vez de programação imperativa;
4. Em Python, não precisamos criar classes, ao contrário de Java;

*Wikipedia: Na Ciência da Computação, programação imperativa é um paradigma de programação que descreve a computação como ações, enunciados ou comandos que mudam o estado (variáveis) de um programa. Muito parecido com o comportamento imperativo das linguagens naturais que expressam ordens, programas imperativos são uma sequência de comandos para o computador executar. O nome do paradigma, Imperativo, está ligado ao tempo verbal imperativo, onde o programador diz ao computador: faça isso, depois isso, depois aquilo... Este paradigma de programação se destaca pela simplicidade, uma vez que todo ser humano, ao se programar, o faz imperativamente, baseado na ideia de ações e estados, quase como um programa de computador.*

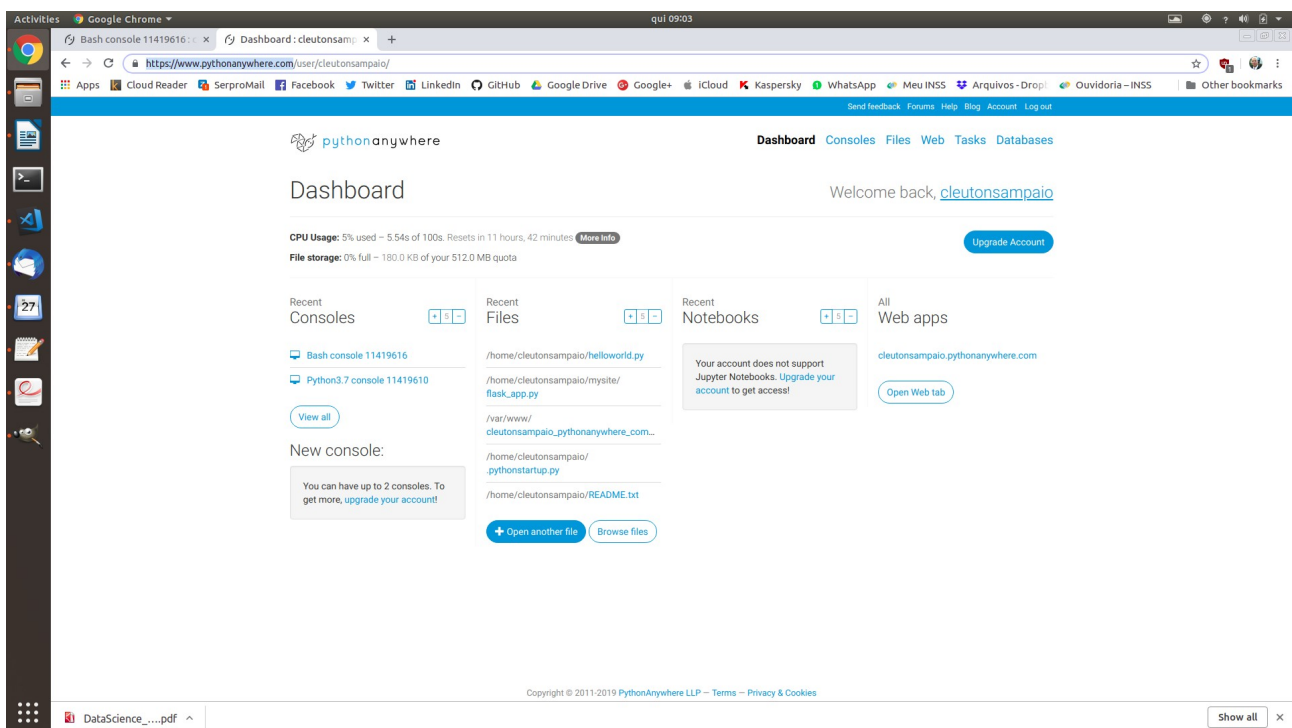
## Plataforma para rodar Python

Me responda uma coisa: Você quer aprender Python, certo? Então, não quer ficar perdendo tempo configurando ambiente. Parece óbvio, mas muitos cursos de programação perdem tempo te ensinando bobagens desnecessárias.

Você vai aprender Python utilizando um site online. Não precisará instalar porcaria alguma em seu computador e, na verdade, nem precisará de um computador, podendo utilizar: Tablet, Smartphone ou até SmartTV. E, como este curso é realizado em laboratórios, nem sempre temos máquinas com capacidade para executar nada além de um navegador.

Existem alguns provedores de Python As A Service, como o site:

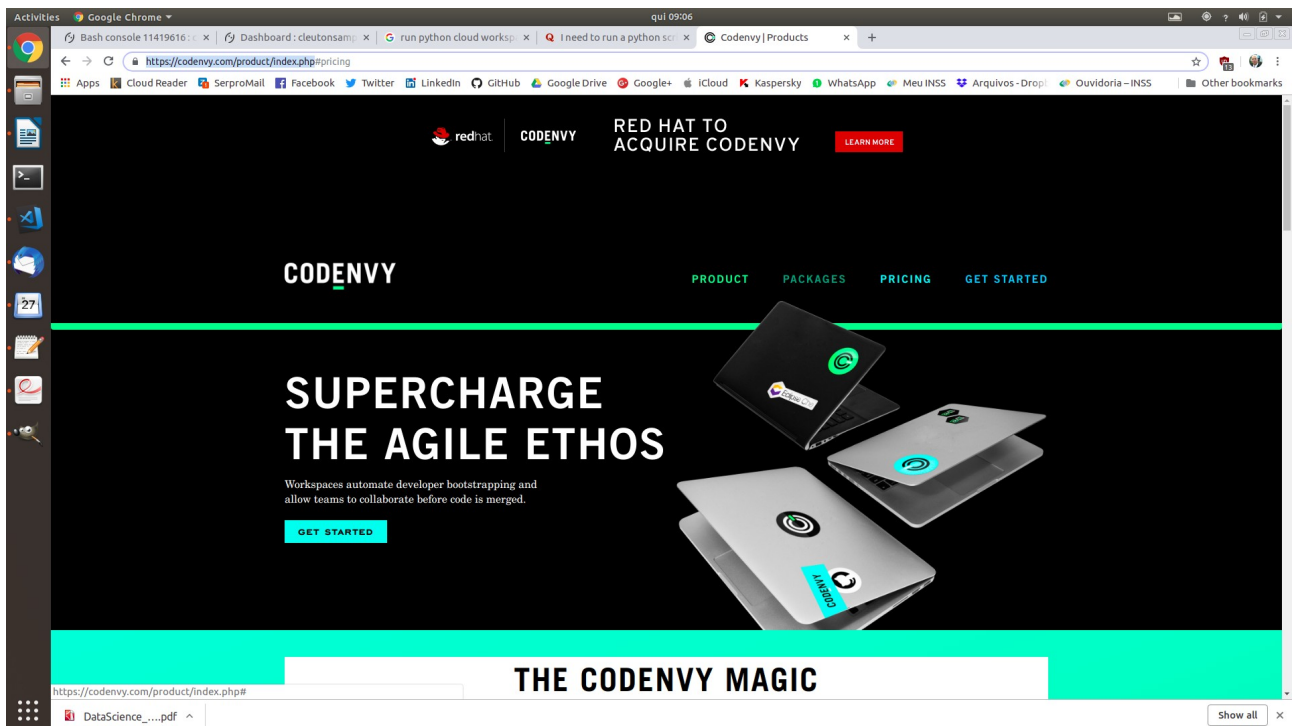
<https://www.pythonanywhere.com>



No PythonAnywhere você pode criar, subir e executar arquivos Python a partir de um navegador, e possui um nível gratuito que lhe permitirá acompanhar todo esse curso.

Outro site interessante é o Codenvy:

<https://codenvy.com/product/index.php>

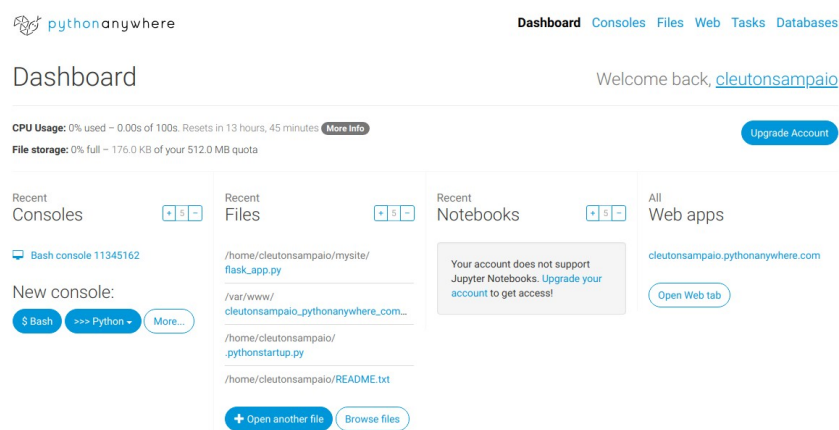


O Codenvy também possui um nível gratuito que lhe permite criar um ambiente Python totalmente em nuvem.

Neste curso, pretendo utilizar o PythonAnywhere, mas vou mostrar como usar ambos.

## PythonAnywhere

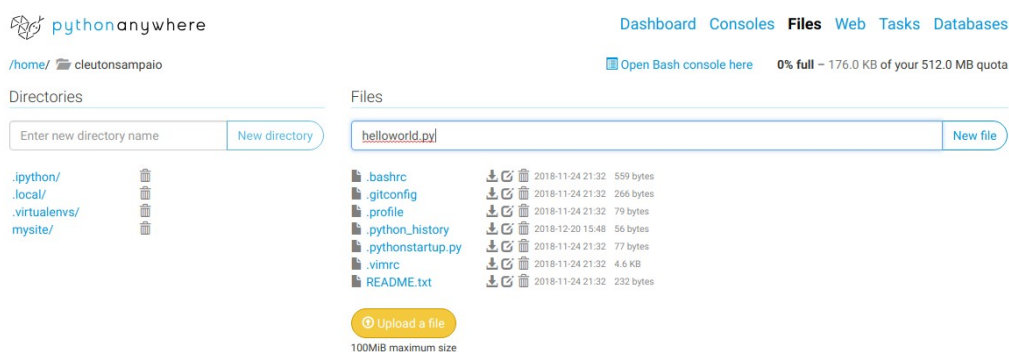
A primeira coisa que você tem que fazer é criar uma conta, o que lhe dará direito a utilizar: Arquivos e consoles.



Aqui você tem acesso aos quatro tipos de elementos que pode interagir:

- Consoles: Para ter acesso ao Python interativo ou a uma console Bash (Linux);
- Files: Para criar, subir ou gerenciar os seus arquivos;
- Notebooks: (só para conta paga) Onde você pode gerenciar seus notebooks Jupyter;
- Web apps: (só uma para conta gratuita) Onde você gerencia sua app web.

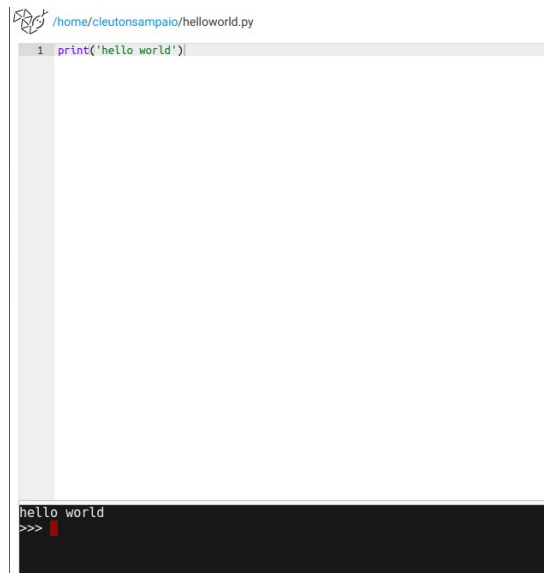
Vamos criar um script Python e executá-lo. Primeiramente, vamos criar um novo arquivo, selecionando a opção “Files” no dashboard:



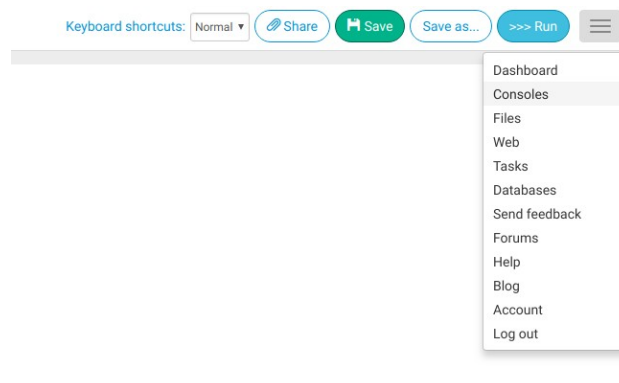
Você pode criar um novo arquivo, como eu fiz na imagem, ou fazer “upload” de um arquivo pronto do seu computador. Ao clicar no botão “New file” é aberto um editor para mim:



Eu escrevi uma só linha, que é um comando “print”. Ele vai mostrar a mensagem “hello world” na console, ou para quem entende Linux: na STDOUT. Note que há alguns botões no alto à direita: Share, Save, Save As e Run. É óbvio o que eles fazem. Eu posso salvar com o Save e depois executar com o Run:



Ai está: O programa foi executado pelo Interpretador Python ou Python Interativo. Depois, é só clicar no ícone de menu e voltar para o dashboard ou para o editor de consoles:



No editor de consoles, podemos criar uma console Python interativa, onde podemos executar comandos imediatos Python (ou até scripts), ou podemos criar uma console Bash, onde fica mais fácil executar scripts:

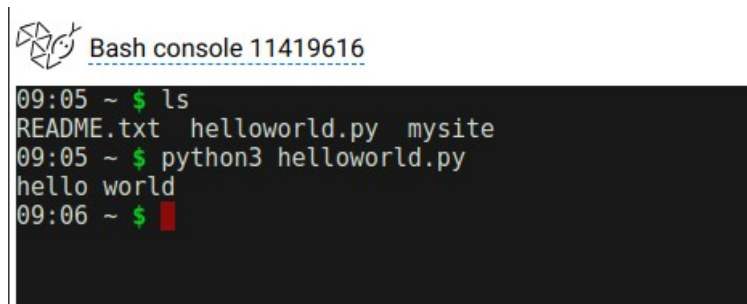
Start a new console:

Python: [3.7](#) / [3.6](#) / [3.5](#) / [3.4](#) / [2.7](#) IPython: [3.7](#) / [3.6](#) / [3.5](#) / [3.4](#) / [2.7](#) PyPy: [2.7](#)  
Other: [Bash](#) | [MySQL](#)  
Custom: [+](#)

Your consoles:

[Python3.7 console 11419610](#) ✕

Ao selecionar “bash” uma console linux é aberta para você e pode executar seu arquivo:

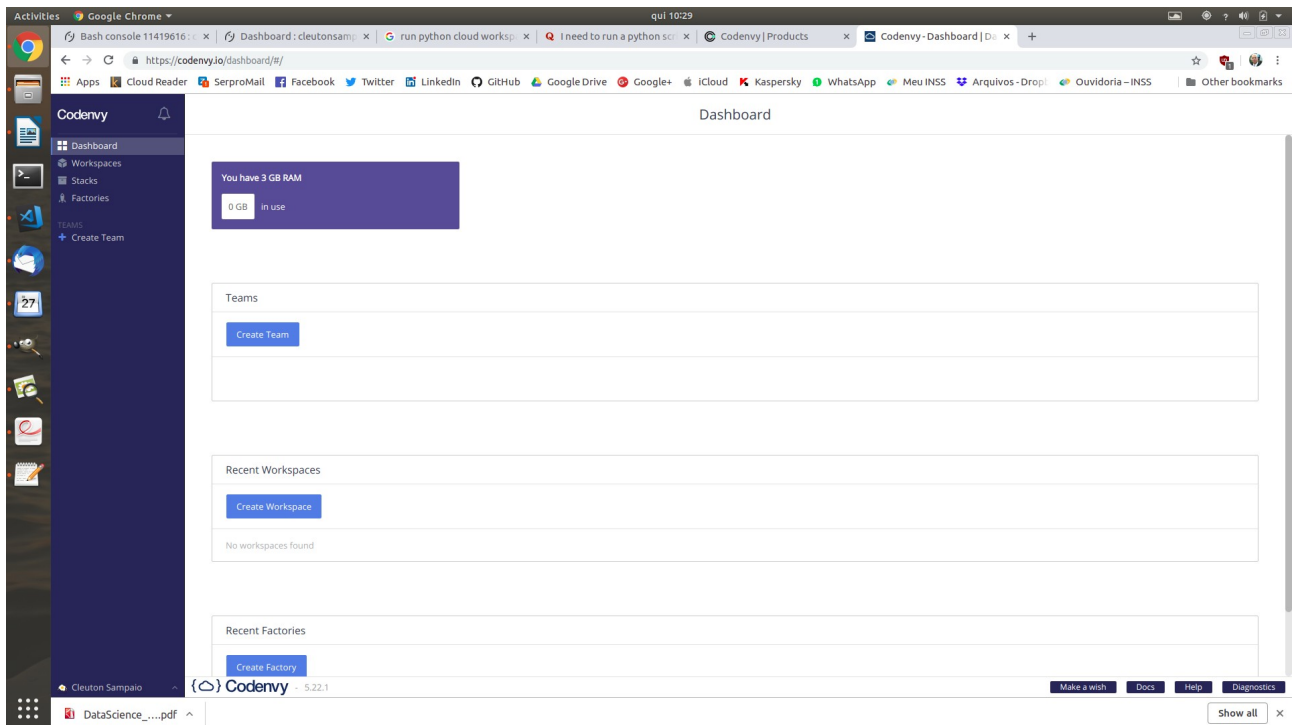


```
09:05 ~ $ ls
README.txt helloworld.py mysite
09:05 ~ $ python3 helloworld.py
hello world
09:06 ~ $
```

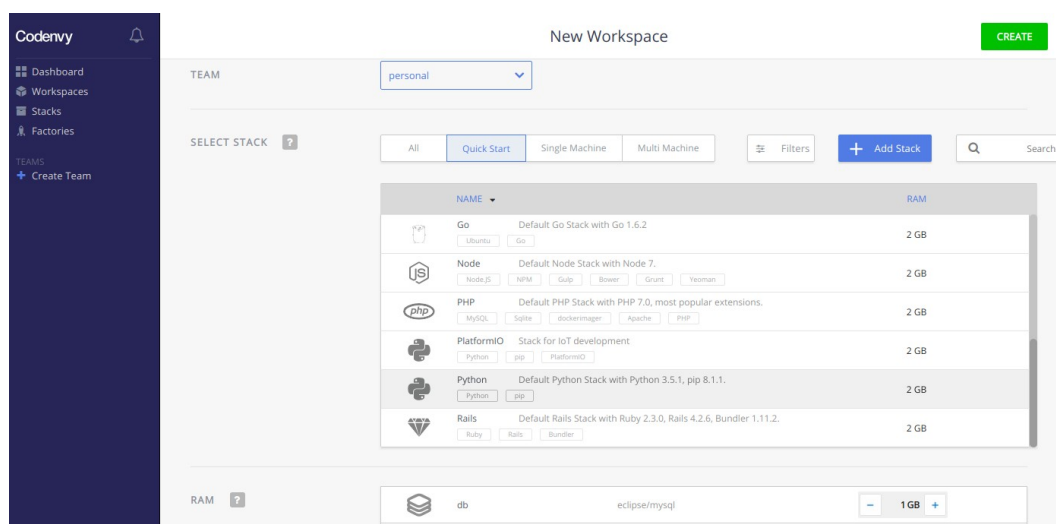
Rodei um comando “ls” e vi que o meu arquivo está lá (“helloworld.py”), portanto, posso usar o comando “python” para executá-lo.

## Codenvy

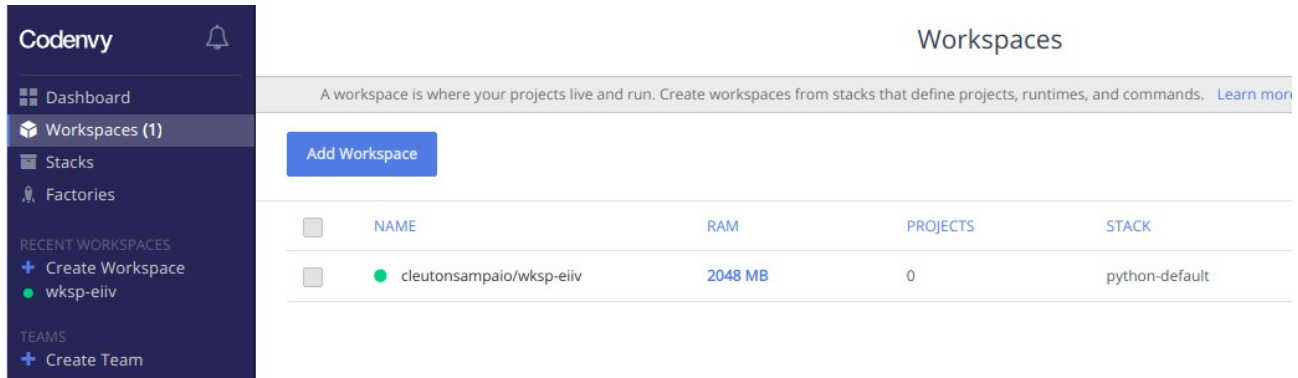
Crie uma conta no site Codenvy ou então use sua conta no LinkedIn ou Github para se logar nele:



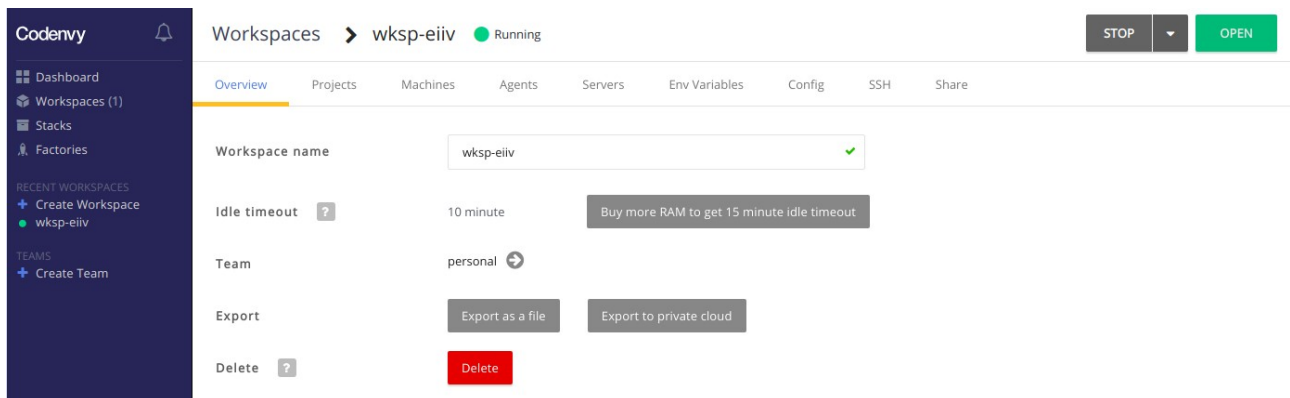
Tudo no Codenvy é feito dentro de uma “workspace”. Você precisa criar uma workspace e adicionar um “Stack”, ou seja, um conjunto de compilador e utilitários:



Selecione o stack “Python” com o Python 3.5.1 e clique no botão “create”. Então, sua workspace será criada:

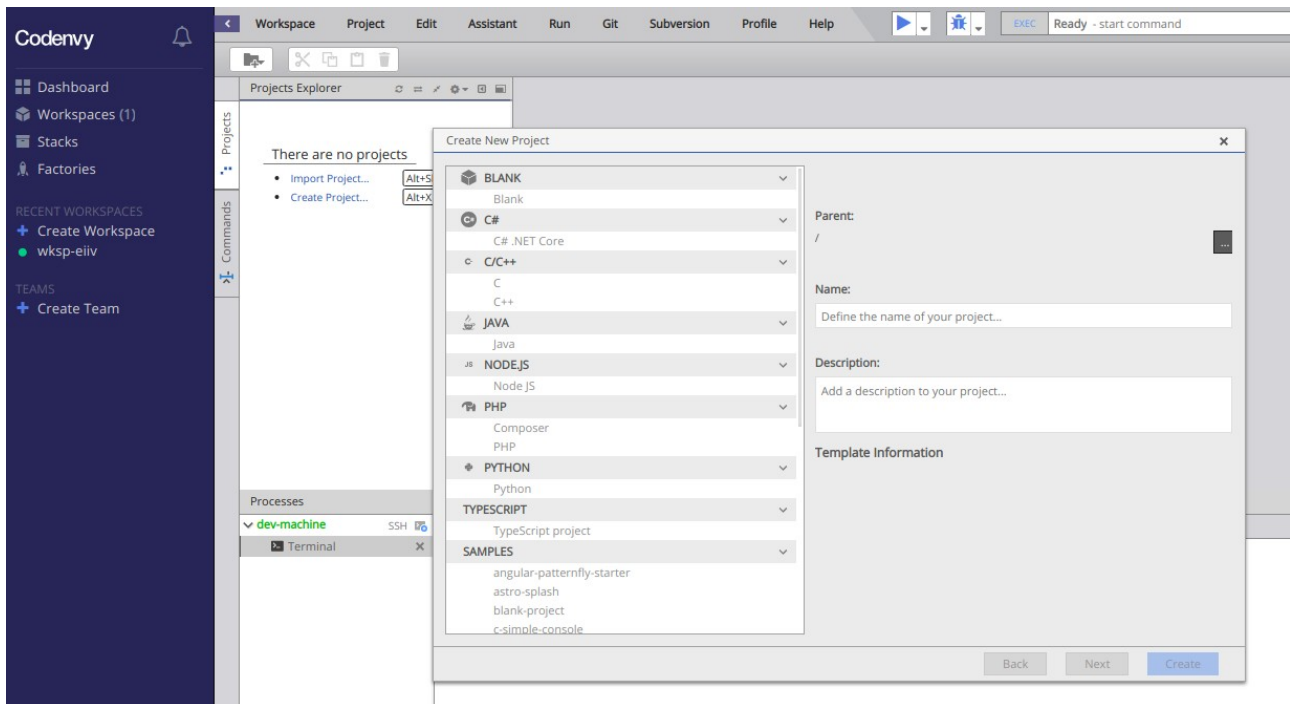


A barra lateral esquerda é um menu, onde suas Workspaces aparecerão. Ao selecionar a opção “workspaces”, vemos a nossa “workspace” criada e sendo executada (botão verde aceso ao lado do nome). Podemos selecionar a workspace:

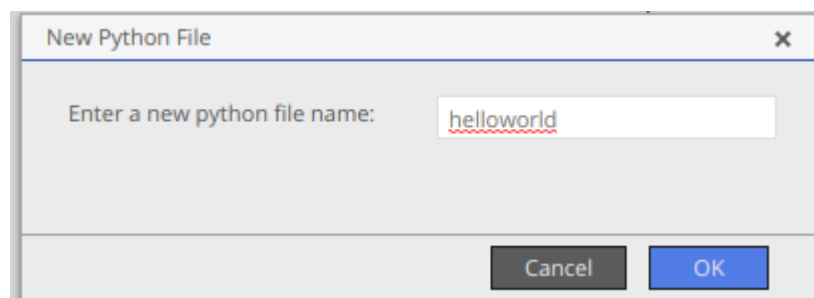


Aqui podemos modificar a workspace ou então abri-la para editar arquivos e executar comandos, bastando clicar no botão “open”. Ao abrir a workspace, você notará que não existe um “projeto” para adicionarmos código, então, deverá clicar em “create project”:

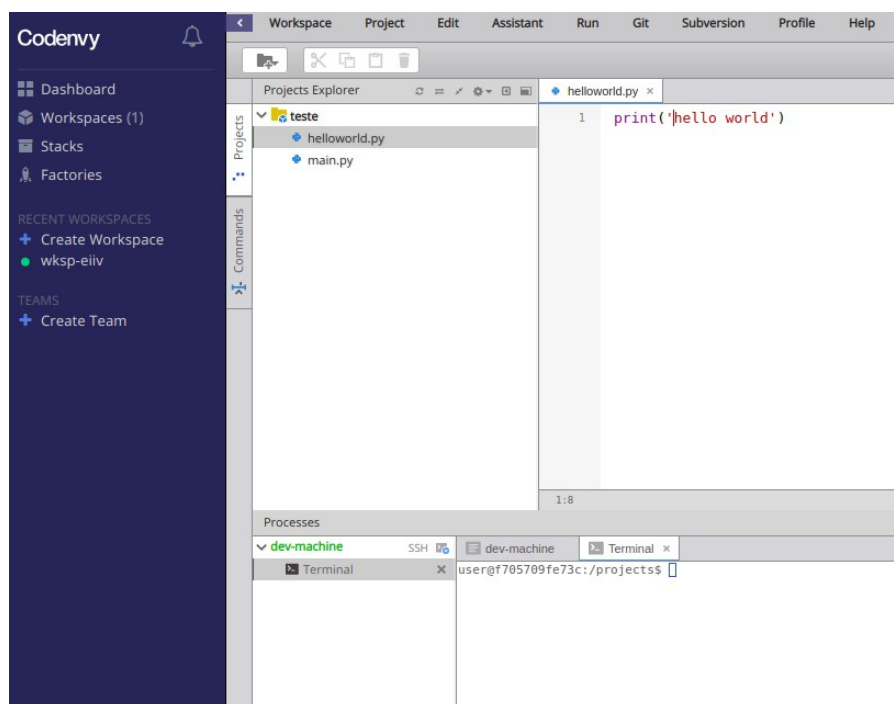




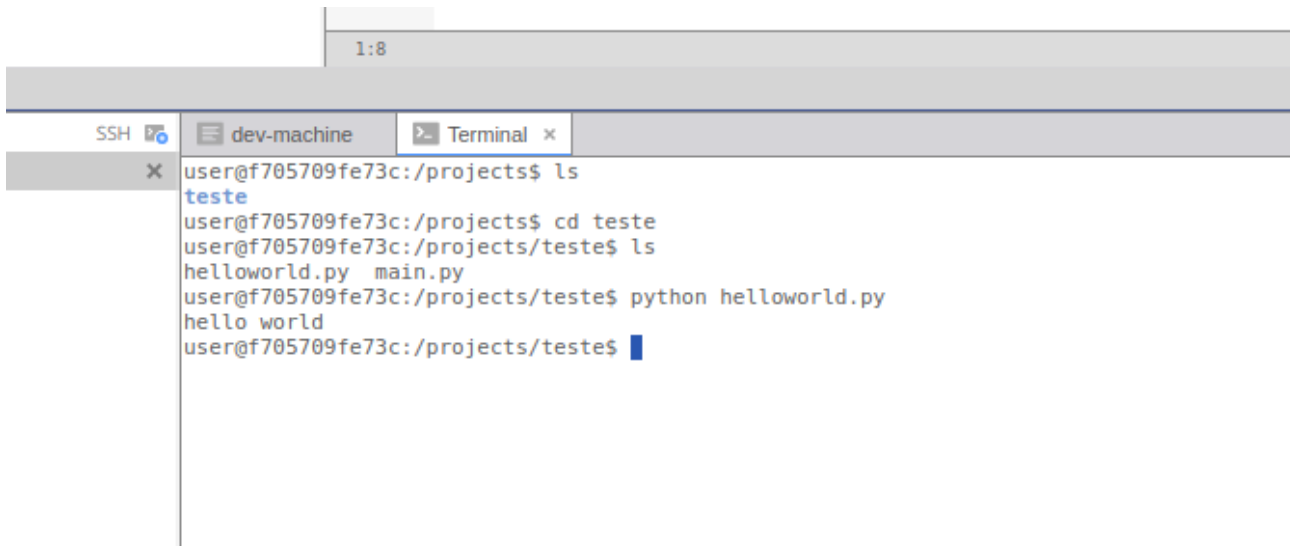
Selecione “python” e informe o nome do Projeto. Então, você poderá adicionar arquivos fonte a ele. Basta selecionar o projeto ir no menu “new” e selecionar “Python file”:



E você já pode editar seu arquivo:



Se quiser executar o script que você acabou de editar, basta ir na janela “terminal”, na parte de baixo da workspace, entrar na pasta do projeto e executar o Python:



The screenshot shows a terminal window titled "dev-machine" with a "Terminal" tab. The prompt is "user@f705709fe73c:/projects\$". The user enters "ls", and the output is "teste". Then, the user enters "cd teste", and the prompt changes to "user@f705709fe73c:/projects/teste\$". The user enters "ls", and the output is "helloworld.py main.py". Finally, the user enters "python helloworld.py", and the output is "hello world".

```
user@f705709fe73c:/projects$ ls
teste
user@f705709fe73c:/projects$ cd teste
user@f705709fe73c:/projects/teste$ ls
helloworld.py  main.py
user@f705709fe73c:/projects/teste$ python helloworld.py
hello world
user@f705709fe73c:/projects/teste$
```

Como eu criei um projeto chamado “teste”, há uma pasta com o mesmo nome, dentro de “projects”, e o meu arquivo “helloworld.py” está lá, juntamente com um “main.py” que ele mesmo criou. Basta executar o nosso arquivo.

## No seu computador

Embora eu aconselhe fortemente a usar o PythonAnywhere ou o Codenvy, sempre existem aquelas pessoas teimosas, que insistem em instalar o Python localmente. Se você é uma delas, então vou mostrar como fazer isso no Linux e no Windows.

### Windows

Baixe o instalador para windows diretamente do site:

<https://www.python.org/downloads/release/python-372/>

Você precisa saber a arquitetura do seu computador e sistema operacional. Se for 64 bits, então selecione a opção: “Windows x86-64 executable installer”. Agora, se for 32 bits, use esta outra opção: “Windows x86 executable installer”.

Execute o instalador e pronto! Você já tem o Python. A instalação é simples, mas eu recomendo você olhar as instruções:

<https://docs.python.org/3.7/using/windows.html>

O Python vem com uma IDE simples chamada “idle”.

## Linux

Os sistemas Linux já vêm com Python instalado. Se for um Linux antigo, virá com Python 2.7. O Ubuntu 18 já vem com Python 3 instalado. Utilizando o comando “python3” podemos executar scripts.

Se o seu Linux não contém Python 3, então ele pode ser instalado assim:

- **Ubuntu** (14.04 e 16.04):

```
$ sudo apt-get update
$ sudo apt-get install build-essential libpq-dev libssl-dev openssl libffi-dev
zlib1g-dev
$ sudo apt-get install python3-pip python3-dev
```

- **Fedora:**

```
sudo dnf install python37
```

## Exercício

Bom, você já sabe como usar Python, seja na nuvem ou no seu computador. Então, pegue o programa que mostrei no tópico “O “jeitão” do Python”, digite e execute. Faça-o de preferência no PythonAnywhere, que é o ambiente utilizado neste curso, mas pode usar no seu desktop também.