

MC102 – Algoritmos e Programação de Computadores



Tarefa de laboratório 00

O objetivo deste exercício é propiciar a familiarização com o interpretador Python e a ferramenta de submissão e testes automáticos SuSy. Teste as opções corretas e todos os erros documentados. Isso facilitará muito o seu processo de depuração durante o desenvolvimento das próximas tarefas.

Hello, world!

Em muitos tutoriais, o primeiro programa ensinado é o que escreve a mensagem *Hello, world!*. Seguiremos nesta linha e escreveremos **Oi, mundo!**. Veja o código abaixo:

```
print("Oi, mundo!")
```

Teste utilizando a shell python3

Abra um terminal e shell python3:

```
$ python3
Python 3.6.4 (default, Feb  8 2018, 14:42:51)
[GCC 7.3.1 20180130 (Red Hat 7.3.1-2)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

A partir deste ponto, você pode testar comandos em Python de forma interativa. Vamos lá!

```
>>> print("Oi, mundo!")
Oi, mundo!
```

Muitas vezes não escrevemos exatamente como o interpretador esperava e temos como retorno uma mensagem de erro bem amigável.

```
>>> print "Oi, mundo!"
File "", line 1
    print "Oi, mundo!"
    ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print("Oi, mundo!")?
```

Às vezes, a mensagem pode ser um pouco mais difícil de entender. Abaixo, interpretador está reclamando que a linha acabou (EOL = end of line) enquanto tentava processar a string. O que está

faltando?

```
>>> print ("Oi, mundo!")
File "", line 1
    print ("Oi, mundo!")
    ^
SyntaxError: EOL while scanning string literal
```

Um erro semelhante leva a uma mensagem bem diferente. Neste caso, como não havia as aspas demarcando o início da string, o interpretador tentou avaliar o conteúdo da string como elementos da linguagem e reclamou que a sintaxe estava errada.

```
>>> print (Oi, mundo!)
File "", line 1
    print (Oi, mundo!)
    ^
SyntaxError: invalid syntax
```

Outro erro comum é digitarmos o nome errado de uma função.

```
>>> prit ("Oi, mundo!")
Traceback (most recent call last):
  File "", line 1, in
NameError: name 'prit' is not defined
>>>
```

Às vezes o nome está correto, mas o interpretador precisa de ajuda para encontrar a função. Veremos isso mais para a frente na disciplina.

Testando via arquivo main.py

Outra forma de usar a linguagem Python é criarmos um arquivo com o programa e para execução posterior. Abra um editor de textos como o `gedit` e crie um arquivo com o conteúdo

```
print("Oi, mundo!")
```

e depois execute da seguinte forma:

```
$ python3 main.py
Oi, mundo!
```

Altere o arquivo introduzindo erros semelhantes aos anteriores e observe as mensagens retornadas.

Testando com o SuSy

No SuSy, para cada tarefa, criamos um conjunto de testes com arquivos de entrada `arq1.in`, `arq2.in`, `arq3.in` e para cada um deles temos uma saída esperada `arq1.res`, `arq2.res`, `arq3.res`. No caso deste exercício teremos apenas um arquivo de teste `arq1.in` com entrada vazia e o resultado deve ser:

```
Oi, mundo!
```

Se você submeter arquivos incorretos como exercitados na seção anterior, receberá respostas parecidas. Agora, observe os comandos abaixo:

```
print("Hello, world")
print("Hallo, Welt!")
print("Oi, mundo!!!")
print("Oiiii, mundooo!")
```

Todos executam sem emissão de mensagem de erro pelo interpretador e cumprem a missão de cumprimentar nosso mundo. No entanto, não seguem exatamente a especificação inicial do enunciado e implicarão em um resultado **incorreto**.

Para testar, submeta `print("Oi, mundo!!!")` e observe o resultado:

Teste 1: resultado incorreto

```
lcl
< Oi, mundo!!!
---
> Oi, mundo!
```

Um pouco mais desafiador é encontrar erros causados por espaços em branco extras ou faltantes. Submeta `print("Oi, mundo! ")` e observe o retorno:

Teste 1: resultado incorreto

```
lcl
< Oi, mundo!
---
> Oi, mundo!
```

Orientações para submissão

Veja [aqui](#) a página de submissão da tarefa. Lembre-se que o arquivo a ser submetido deve se chamar **main.py**.

Todos os alunos matriculados até 2 de março estão inscritos. Se você entrou depois e/ou não estiver cadastrado corretamente, envie email para islene@ic.unicamp.br.

Observe o limite máximo de 10 submissões e que a última submissão deve ser correta.

O peso desta tarefa é 1.

O prazo final para submissão é 10/03/2018.
