Processamento da Informação

Fabrício Olivetti de França

02 de Fevereiro de 2019

UFABC

CMCC
Centro de Matemática, Computação e Cognição
Center for Mathematics, Computation and Cognition

Topics

1. Processamento da Informação

Processamento da Informação

Primeira tarefa da disciplina:

O que é programação de computadores?

Programação de Computadores

Construir e implementar um **código executável** ou **interpretável** para fazer alguma tarefa.

Programas de Computador

Quais exemplos de programas de computador vocês conseguem imaginar?

Programas de Computador

- Windows?
- App do Facebook para celular?
- Filtros do Instagram?
- Jogador inteligente de xadrez?
- Interface com uma TV?

Passos para construção de um programa

1. Entender o problema
2. Entender **a solução** para o problema
3. Sistematizar a solução para o problema
4. Escrever a solução de forma que o computador entenda
5. Testar solução
6. Analisar o desempenho do programa

etc.

Passos para construção de um programa

Nessa disciplina o foco será nos 3 primeiros itens!

Algoritmo e Computação

Um **algoritmo** é uma descrição precisa e desambígua de como resolver um problema.

Computação é o cálculo feito seguindo a descrição do algoritmo.

Algoritmo

Um algoritmo em sua forma mais primitiva pode ser escrito como uma função matemática.

Algoritmo

Um algoritmo que divide um valor por 10

```
f(x) = x/10
```

Algoritmo

Qual o resultado para:

- $f(1)$
- $f(12)$
- $f(123)$
- $f(1234)$

Algoritmo

Um algoritmo que calcula o resto da divisão por 10

```
g(x) = x % 10
```

Algoritmo

Qual o resultado para:

- $g(1)$
- $g(12)$
- $g(123)$
- $g(1234)$

Quando a diversão começa?

O segredo da construção de algoritmos mais interessantes é a **composição**.

A ideia geral é começar construindo funções pequenas e simples e, depois, utilizá-las para programas mais complexos!

Algoritmo

Qual o resultado para:

- $g(f(1234))$
- $f(g(1234))$

Avaliação da parte prática

Nessa turma de prática faremos algo um pouco diferente do habitual:

- Na segunda metade de todas as aulas faremos uma atividade valendo nota (1.5 pto).
- No início de cada atividade, o aluno receberá uma letra correspondente a entrada de seu programa. NÃO UTILIZE OUTRA LETRA!
- Após a conclusão da atividade, o programa deve emitir uma (ou mais) respostas que devem ser escritas na folha de resposta e entregue ao professor ao final da aula.

Utilizaremos uma única folha de resposta para todas as atividades!

Folha de resposta

Proibições:

- NÃO leve a folha de resposta para casa.
- NÃO rasure a letra de entrada de seu programa ou o visto do professor.

O não cumprimento de alguma dessas regras acarretará em reprovação automática!

Multa por atraso

Caso o aluno não consiga completar o programa ou entregar a atividade com a resposta incorreta, poderá refazer a atividade em casa e preencher a resposta na folha de atividade até duas semanas após. A nota da atividade atrasada será de 0.5 ponto.

Para isso a folha terá um campo apropriado para respostas em atraso.

Nota final

A nota final para o laboratório será a soma das 7 maiores notas limitado em 10.

Caso o aluno falte em mais do que 3 atividades será reprovado por falta.

Programa de Computador

Um programa de computador é um conjunto de instruções de máquina que implementam um algoritmo.

Passo a passo de como o computador deve processar os dados.

Programa de Computador

Internamente ele é definido por sequências de bits.

Cada sequência de bit é mapeada para uma instrução do processador.

Programa de Computador

001000	00001	0000000101011110
Código OP	Endereço	Valor
add	eax	360

Linguagem de Máquina

- Difícil de ler, escrever e entender.
- Requer uma lógica de programação imperativa.

Linguagem de Alto Nível

Para resolver esses problemas, foram criadas linguagens de programação que serviriam como intermediários entre a linguagem de máquina e o programador.

Linguagem de Alto Nível

- Possui um conjunto de instruções próximas da linguagem natural.
- Minimiza o número de instruções para tarefas frequentes.
- Não requer completo entendimento do funcionamento do computador.

Linguagem Python

Nesse curso utilizaremos a linguagem de programação **Python**:

- Tem uma sintaxe simples.
- Filosofia de códigos pequenos e legíveis.
- Declarativo (ou quase).

Linguagem de Máquina

Nas aulas práticas construiremos um interpretador de linguagem de máquina!

Instalando o Python

Acesse o site:

https://www.anaconda.com/distribution/

Instalando o Python

Clique em *Download* e, então, clique no botão *Download* logo abaixo de *Python 3.7 version*.

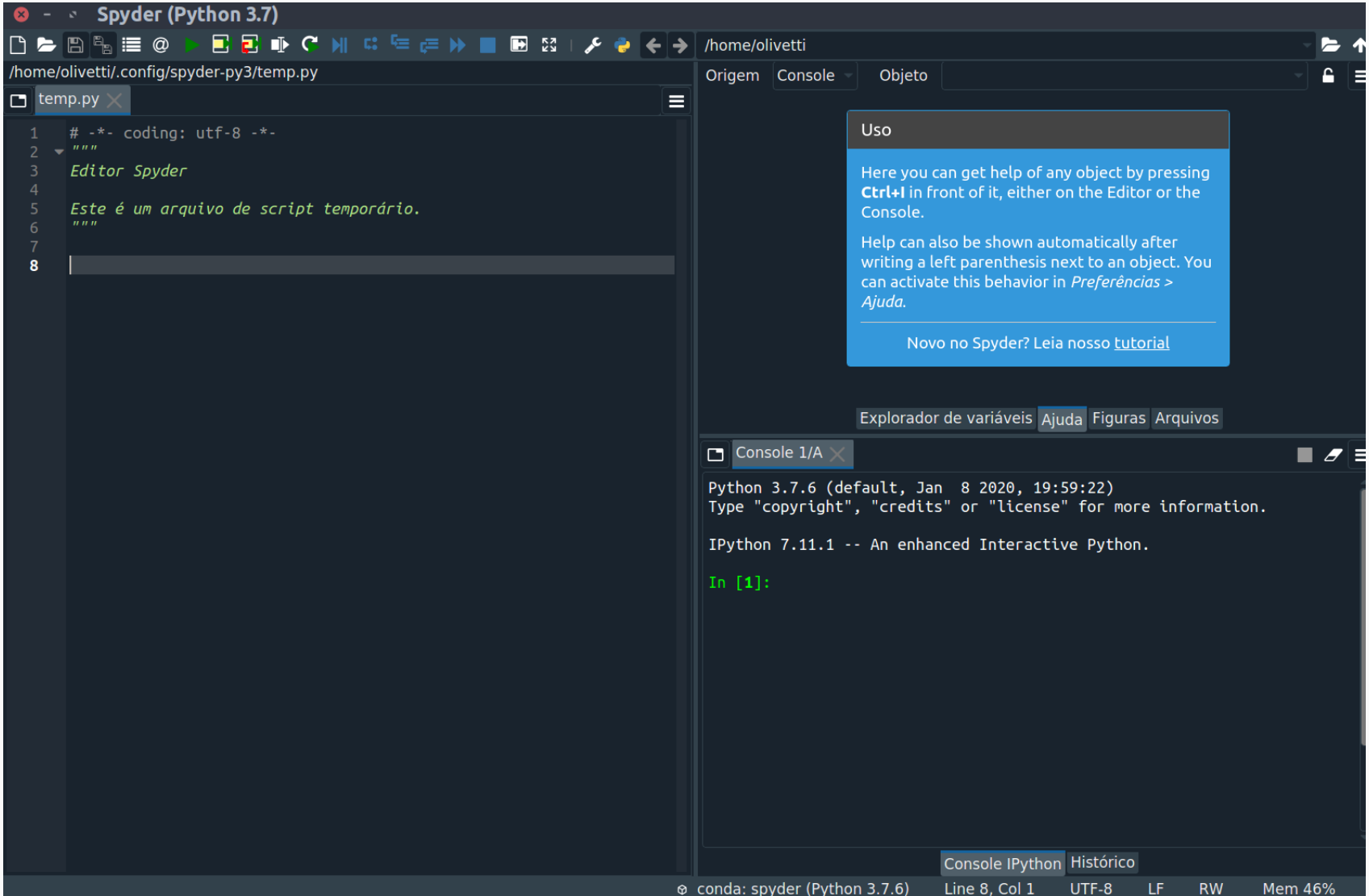
Siga os passos de instalação!

Instalando o editor

Abra o *Anaconda Navigator* procure por *Spyder* e clique em *Install*.

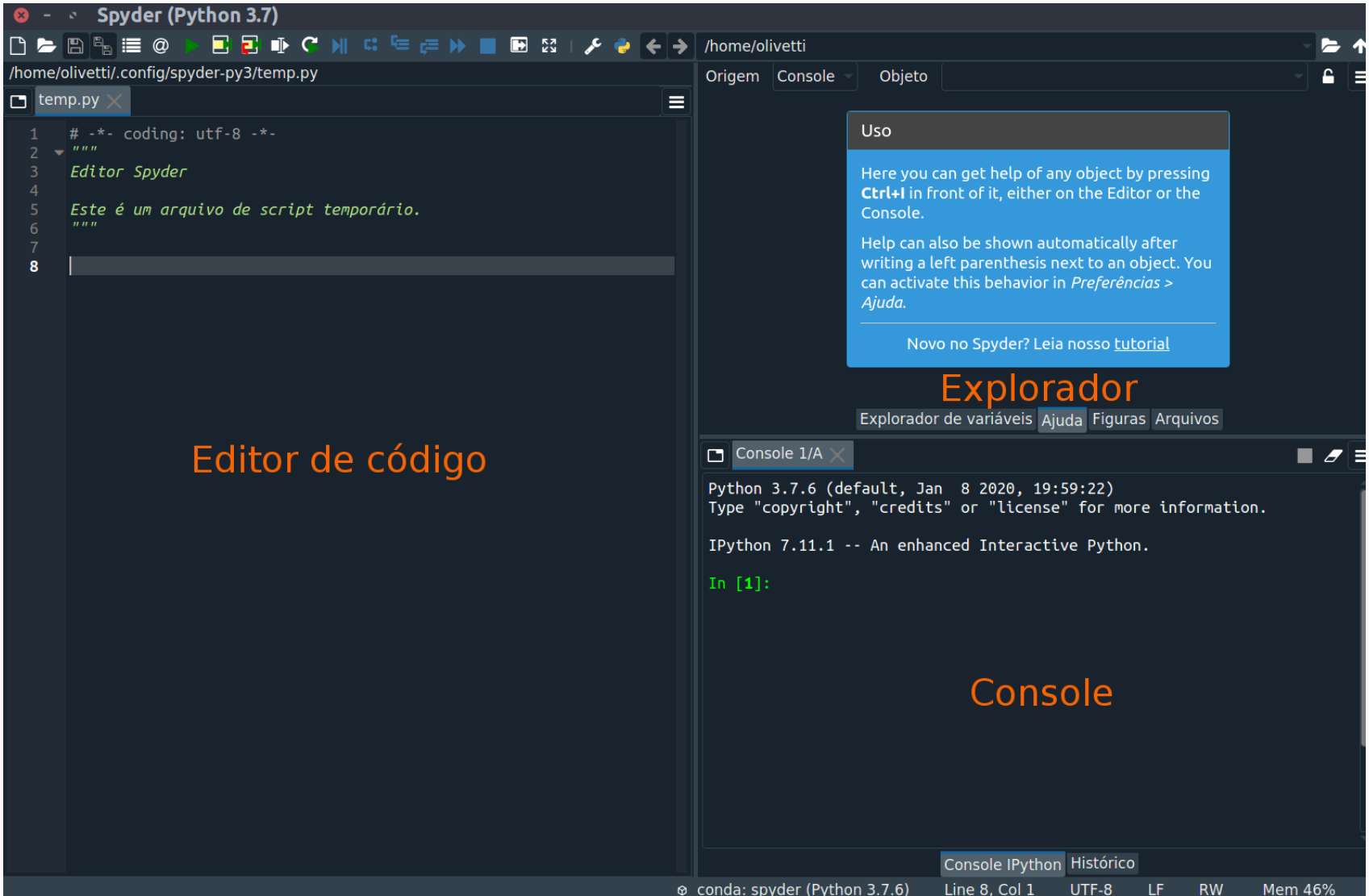
Testando o editor

Abra o editor clicando em *Launch*. Deve aparecer uma tela parecida com essa:



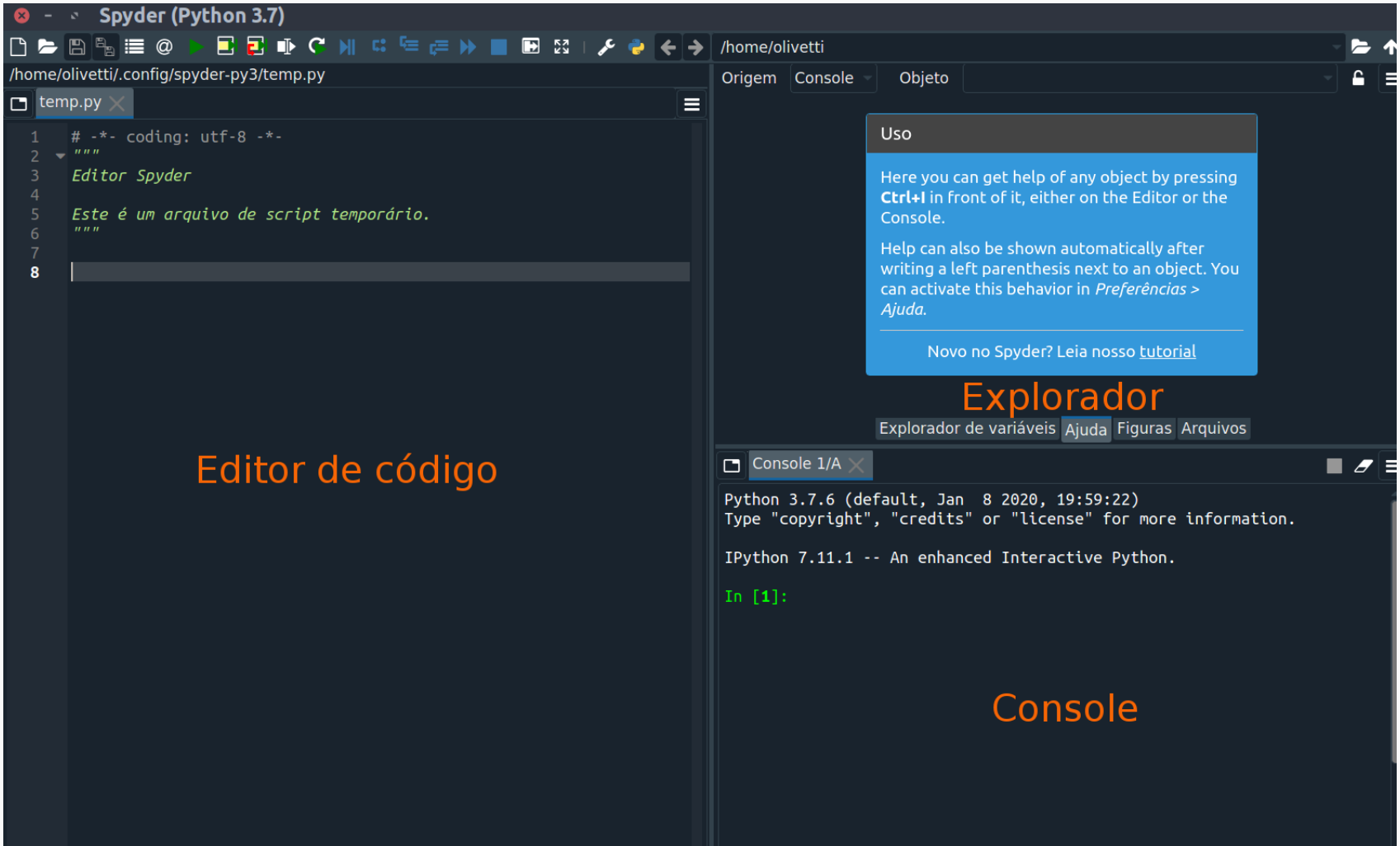
Testando o editor

A janela da esquerda é chamada de **Editor**. É nela que você escreverá seus códigos.



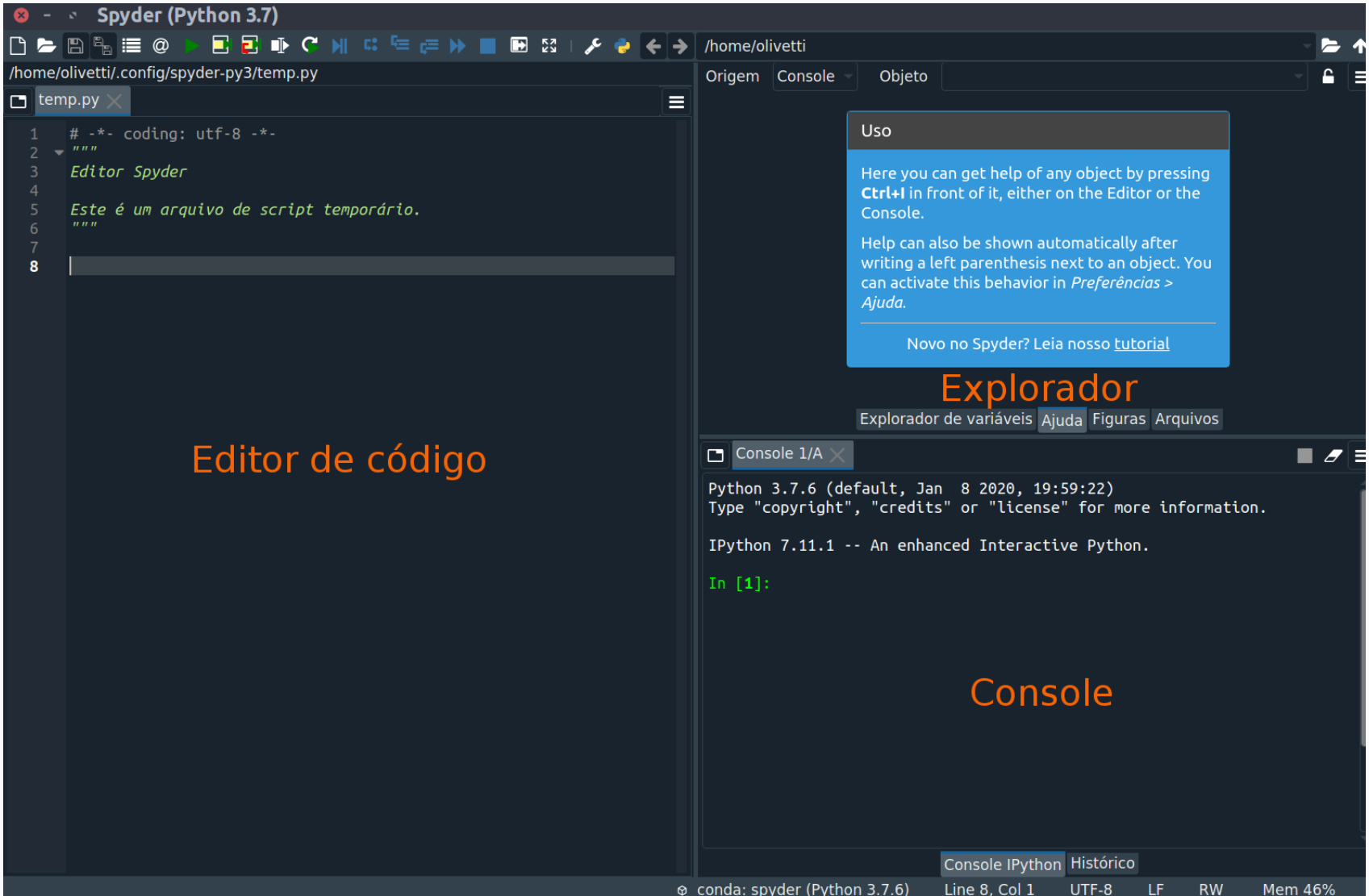
Testando o editor

A janela no canto superior direito é o **Explorador**. Ela te traz informações sobre o estado atual do seu programa, quando em execução.



Testando o editor

A janela no canto superior direito é o **Console**. Ele te permite fazer alguns testes pontuais ou executar programas.



Tarefa

No **Console** escreva $1 + 2$ e pressione a tecla *enter*. Qual o resultado?

Tarefa

Repita agora escrevendo a expressão $1 + 2 * 3$. Tente advinhar o resultado antes de apertar *enter*.

Tarefa

Repita agora escrevendo a expressão $(1 + 2) * 3$. Tente advinhar o resultado antes de apertar *enter*.

Precedência

Em operações aritméticas a linguagem de programação define ordem de precedência para calcular as operações. Primeiro ele calcula qualquer cálculo de potência, em seguida multiplicações e divisões e, finalmente, somas e subtrações.

Pergunta

Em uma expressão $a + b - c$ faz diferença calcular a subtração antes da soma?

Pergunta

Em uma expressão $a * b/c$ faz diferença calcular a divisão antes da multiplicação?

Tarefa

Calcule agora 3/2, qual o resultado? E se fizermos 3//2?

Operações com Inteiros

Por convenção, utilizamos `/` para um resultado nos números reais e `//` para resultados inteiros.

Criando e manipulando variáveis

Digite agora no console o seguinte comando:

```
x = 2
```

Criando e manipulando variáveis

Na janela *Explorador* clique em *Explorador de Variáveis*. Essa janela passa a mostrar o tipo, tamanho e conteúdo de nossa variável `x`.

Vocês estão de acordo com o que está sendo mostrado?

Tarefa

Teste agora:

```
x = 2.3
x = "Ola mundo!"
x = [1,2,3,4]
```

E verifique como os valores do explorador se alteram.

Variáveis

Uma variável armazena um ou mais valores de um certo tipo de dado atribuindo um nome para esse(s) valor(es).

O que podemos fazer com uma variável?

Podemos alterar seu valor, fazer operações, mostrar seu valor, utilizar como argumentos de uma função, etc.

Tarefa

Faça a seguinte sequência de instruções, o que aparece no *Explorador de Variáveis*?

```
x = 2
y = 3
z = x+y
```

Tarefa

Faça:

```
x = 2
y = "ola"
z = x+y
```

O que acontece?

Tarefa

Faça agora:

```
print(x,y,z)
```

O que aparece no console?

Imprimindo valores

A função `print` recebe um ou mais argumentos de qualquer tipo e mostra os valores na tela:

```
print("Ola mundo")
print(1+2)
print(x)
print(x, " + ", y, " = ", z)
```

Tarefa

Para tornar essa última instrução `print` mais fácil de escrever, podemos usar as *strings de formatação*:

```
print(f"{x} + {y} = {z}")
```

A *string de formatação* inicia com a letra *f* seguido do que quer que apareça na tela entre aspas duplas e toda variável entre *chaves*.

Editor

Para que serve o editor? Quando queremos criar uma sequência de instruções, escrever programas maiores, organizar e guardar nosso programa.

Tarefa

Escreva no editor:

```
x = 10
y = 20
z = x+y
print(f"{x} + {y} = {z}")
```

Clique no ícone de triângulo verde (ou aperte Ctrl+Enter).
Esse botão tem a função de **executar** o programa.

Nossa primeira função

Para definir uma função, fazemos:

```
def funcao(x):
    return x+1
```

Nossa primeira função

O termo `def` indica que definirei uma nova função, em seguida atribuímos um nome para ela e, entre parênteses, as variáveis de entrada para a função, terminando a linha com “:”.

Nossa primeira função

Na linha seguinte, alinhamos (utilizando espaço) o código 4 colunas a direita e escrevemos a instrução a ser executada.

Nossa primeira função

A palavra `return` indica que a instrução nessa linha é o que a função retorna como saída.

Tarefa

Escreva a função de exemplo no editor, clique em executar e digite no console:

```
funcao(1)
```

O resultado foi o que você esperava?

Tarefa

Vamos criar nossas duas funções de exemplo do início da aula:

```
def div10(x):
    return x//10

def mod10(x):
    return x%10
```

Tarefa

Qual o resultado de:

```
div10(1234)
```

```
mod10(1234)
```

O que essas funções estão fazendo?

Tarefa

Crie uma função para recuperar o penúltimo dígito utilizando `div10` e `mod10`.