



O que vamos aprender nessa aula?

- **Criar nosso primeiro código em Python para:**
 - imprimir dados na tela do computador
 - receber dados do usuário
 - entender o que são os tipos de dados
 - converter tipos de dados
 - utilizar variáveis para armazenar dados
 - instalar e utilizar a biblioteca `fpdf`

Nosso primeiro código em Python

O ambiente Jupyter Notebook

- Células de texto (Markdown)
- Células de código (Code)
- Comentários no Python

Isso é uma célula de texto

```
In [1]: # isso é um comentário no Python
```

Imprimindo dados na tela do computador

- Para imprimir dados na tela do computador, utilizamos a função **print()**.

```
In [2]: print("Semana Python na Prática")  
Semana Python na Prática
```

```
In [3]: print("Meu primeiro código")  
Meu primeiro código
```

Tipos de Dados: textos e números

- Texto: **str**
- Numérico:
 - inteiro: **int**

- com casa decimal: **float**
- **ATENÇÃO:** o separador de casas decimais no Python é o **ponto** e não a vírgula
 - **10,3** (errado)
 - **10.3** (correto)
- Para verificar o tipo de dados, podemos utilizar a função **type()**.
- **Operadores aritméticos:**
 - adição: +
 - subtração: -
 - multiplicação: * (asterisco)
 - divisão: / (barra)

```
In [4]: # isso é um tipo texto
print("10")
```

```
10
```

```
In [5]: type("10")
```

```
Out[5]: str
```

```
In [6]: # isso é um tipo numérico (número inteiro)
print(10)
```

```
10
```

```
In [7]: type(10)
```

```
Out[7]: int
```

```
In [8]: # isso é um tipo numérico (número com casa decimal)
print(10.5)
```

```
10.5
```

```
In [9]: type(10.5)
```

```
Out[9]: float
```

```
In [10]: print(10 + 1)
```

```
11
```

```
In [11]: print(10 - 5)
```

```
5
```

```
In [12]: print(10 * 10)
```

```
100
```

```
In [13]: print(100 / 10)
```

```
10.0
```

```
In [14]: # CUIDADO!!!! Não podemos realizar cálculos com textos e nem entre tipos diferentes!
print("10 + 20")
```

```
10 + 20
```

```
In [15]: # gera um erro, pois estamos somando um texto com um número inteiro!
print("10" + 10)
```

```
-----
TypeError                                Traceback (most recent call last)
Input In [15], in <cell line: 2>()
      1 # gera um erro, pois estamos somando um texto com um número inteiro!
----> 2 print("10" + 10)

TypeError: can only concatenate str (not "int") to str
```

Recebendo dados do usuário

- Para receber dados de um usuário, podemos utilizar a função **input()**.

```
In [16]: input()

Semana do Python na Prática
Out[16]: 'Semana do Python na Prática'
```

```
In [17]: input("Digite o seu nome: ")
         input("Digite a sua idade: ")

Digite o seu nome: Maria
Digite a sua idade: 29
Out[17]: '29'
```

Variáveis: guardando informações

- Variáveis são **espaços na memória do computador** que o Python utiliza para guardar um dado.
- Nome de variáveis não podem:
 - ser palavras reservadas da linguagem
 - começar com um número
 - conter espaço em branco e nem caracteres especiais (acento, cedilha....etc)

```
In [18]: nome = "Semana do Python na Prática"
         print(nome)

Semana do Python na Prática
```

```
In [19]: numero1 = 10
         numero2 = 20
         resultado = numero1 + numero2
         print(resultado)

30
```

```
In [20]: nome = input("Digite o nome: ")
         idade = input("Digite a sua idade: ")
         peso = input("Digite o seu peso: ")

Digite o nome: Maria
Digite a sua idade: 29
Digite o seu peso: 60.5
```

```
In [21]: # CUIDADO! Todos os dados que digitamos no input() são TEXTO, mesmo os números!
         type(nome)

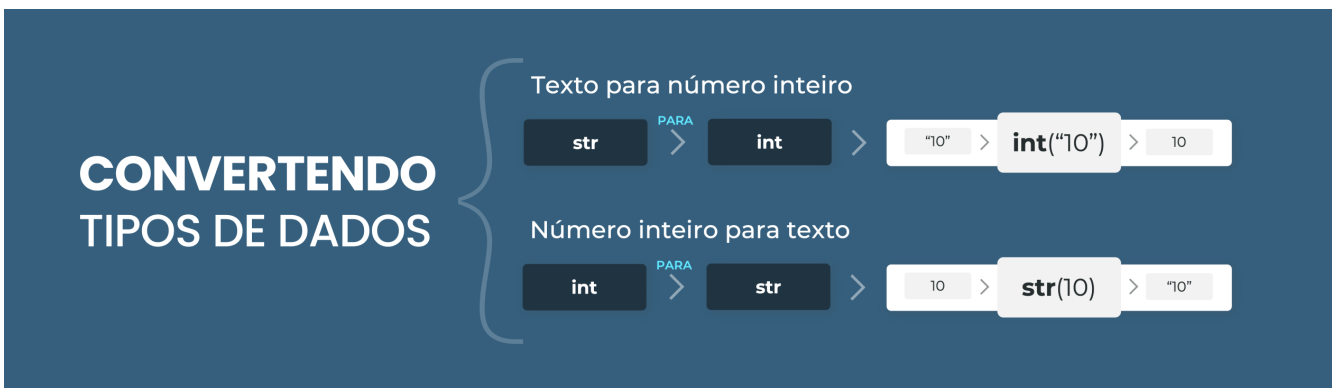
str
Out[21]:
```

```
In [22]: type(idade)

str
Out[22]:
```

```
In [23]: type(peso)
```

```
Out[23]: str
```



- Para converter números em texto, podemos utilizar a função **str**.
- Para converter textos em número, podemos utilizar as funções:
 - **int**: para converter em um número inteiro
 - **float**: para convert em um número com casas decimais

```
In [24]: print(int(idade) + 10)
```

```
39
```

```
In [25]: print(float(peso) + 10)
```

```
70.5
```

```
In [26]: print(float("10.34"))
```

```
10.34
```

```
In [27]: type(10)
```

```
Out[27]: int
```

```
In [28]: type(str(10))
```

```
Out[28]: str
```

Juntando textos com variáveis

- Para juntar textos com os dados de uma variável, podemos utilizar os **textos formatados**.
 - basta utilizar um **f** antes das aspas do texto e colocar a **variável dentro de chaves**.

```
In [29]: nome = input("Digite o seu nome: ")  
idade = input("Digite a sua idade: ")
```

```
Digite o seu nome: Maria  
Digite a sua idade: 29
```

```
In [30]: print(f"Meu nome é {nome} e eu tenho {idade} anos!")
```

```
Meu nome é Maria e eu tenho 29 anos!
```

PROJETO DA AULA

OBJETIVO > Gerar **automaticamente** PDFs com PYTHON

CENÁRIO > Emitir orçamentos para sua empresa



Desenvolvendo as entradas de dados

- Descrição do projeto
- Total de horas estimadas
- Valor da hora de trabalho
- Prazo de entrega estimado

```
In [32]: projeto = input("Digite a descrição do projeto: ")
horas_estimadas = input("Digite o total de horas estimadas: ")
valor_hora = input("Digite o valor da hora trabalhada: ")
prazo_estimado = input("Digite o prazo estimado: ")
```

Digite a descrição do projeto: Desenvolvimento Python
Digite o total de horas estimadas: 60
Digite o valor da hora trabalhada: 120
Digite o prazo estimado: 2 meses

Criando o cálculo do valor total estimado

- **Cálculo:** valor total estimado = total de horas estimadas x valor da hora de trabalho

```
In [33]: valor_total_estimado = int(horas_estimadas) * int(valor_hora)
```

```
In [34]: print(valor_total_estimado)
```

7200

Gerando o PDF com o orçamento

Instalando a biblioteca

```
In [ ]: !pip install fpdf
```

```
In [35]: # importando a biblioteca
from fpdf import FPDF
```

Criando um arquivo PDF

```
In [36]: pdf = FPDF()
pdf.add_page()
pdf.set_font("Arial")
```

Inserindo os dados no PDF

```
In [37]: # utilizando um template
pdf.image("template.png", x=0, y=0 )

# inserindo os dados do projeto
pdf.text(115, 145, projeto)
pdf.text(115, 160, horas_estimadas)
pdf.text(115, 175, valor_hora)
pdf.text(115, 190, prazo_estimado)
pdf.text(115, 205, str(valor_total_estimado))
```

Salvando o arquivo

```
In [39]: pdf.output("orçamento.pdf")
print("Orçamento gerado com sucesso!")
```

Orçamento gerado com sucesso!

PARABÉNS!

Você concluiu o seu primeiro projeto em Python!

E AMANHÃ

Vamos desenvolver um projeto para automatizar tarefas no Python, **você não pode perder!**

