

Ricardo Sérgio

Introdução à Programação e Algoritmia



API B – 12º Ano

1
2
3
4
5
6
7
8
9
10
11
12
13
14

05 {

#Variáveis

}

Variáveis{

1
2
3  Tal como o nome indica é algo **variável** ao longo do nosso **programa**

4  Serve para **guardar informação** necessária á **execução** do **programa**

5
6  Exemplo:

7
8  **Pontuação** de um jogo, **vidas**, **saldo**, etc

9
10
11
12
13 }
14

Variáveis{

- 🐍 O nome das variáveis pode incluir letras, números e underscore
- 🐍 No entanto, nome delas tem de começar por uma letra ou underscore e nunca por um nº
- 🐍 Não podem ter o nome de keywords reservadas ao Python (por exemplo lambda, import, global, etc)
- 🐍 Não podem ter caracteres especiais como +, -, *, /...
- 🐍 São case sensitive logo, por exemplo a variável de nome:
coordenadx ≠ COORDENADAx ≠ Coordenadx ≠ coordenadaX

Variáveis{

- 🐍 Ao contrário de outras linguagens **não necessitam** ser **declaradas previamente** ou sequer **dizer o tipo de dados** que irão **conter**
- 🐍 Passam a **existir** no **momento** em que se **atribui** um **valor**
- 🐍 Podem ser **texto** ou **número** e podem **ser** os **dois** em **momentos diferentes** do código!!!
- 🐍 Grrrrrrrrr já chega. Mas **afinal** o que **são variáveis???**

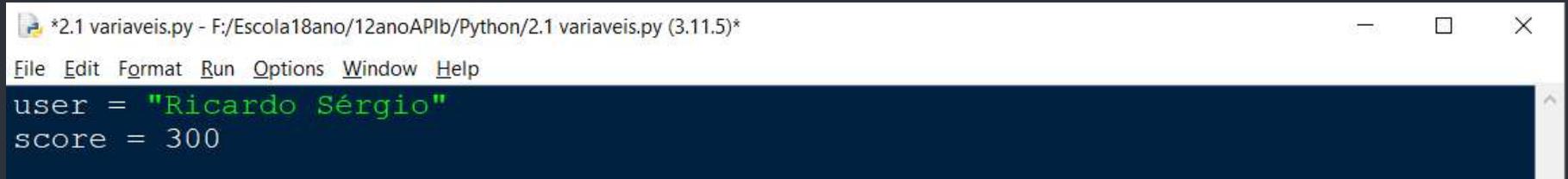
}

Declaração{

 Sintaxe:

nome_variavel = "texto" ou número

Exemplo:



```
*2.1 variaveis.py - F:/Escola18ano/12anoAPIb/Python/2.1 variaveis.py (3.11.5)*
File Edit Format Run Options Window Help
user = "Ricardo Sérgio"
score = 300
```

}

Declaração{

Se a **variável** for **String** (**texto**) pode ser **atribuída** a texto entre **aspas** ou **plicas**

```
*2.1 variaveis.py - F:/Escola18ano/12anoAPIb/Python/2.1 variaveis.py (3.11.5)*
File Edit Format Run Options Window Help
user_1 = "Ricardo Sérgio"
user_2 = 'Joana'
```

Caso seja **número** pode ser
int – inteiro (positivo ou negativo)
float – real (positivo ou negativo)

```
*2.1 variaveis.py - F:/Escola18ano/12anoAPIb/Python/2.1 variaveis.py (3.11.5)*
File Edit Format Run Options Window Help
score = 300
money = 24.5
```

Declaração múltipla{

Podemos declarar várias variáveis em simultâneo

```
2.1 variaveis.py - F:/Escola18ano/12anoAPIb/Python/2.1 variaveis.py (3.11.5)
File Edit Format Run Options Window Help
x, y, z = "Laranja", "Maçã", "Pêra"
```

Ou atribuir mesmo valor a várias variáveis

```
2.1 variaveis.py - F:/Escola18ano/12anoAPIb/Python/2.1 variaveis.py (3.11.5)
File Edit Format Run Options Window Help
a=b=c=10
```

}

Como Imprimir?

Como fazemos para **imprimir** o **conteúdo** da **variável**?

Fácil! Já sabemos que temos de utilizar a instrução `print`, falta saber o que **colocar dentro** do **parênteses**

```
*2.1 variaveis.py - F:/Escola18ano/12anoAPIb/Python/2.1 variaveis.py (3.11.5)*
File Edit Format Run Options Window Help
Score = 300
print ("Score")
```



O **código** acima está **correto**? Pois assim **imprimia** a **palavra Score**

```
*2.1 variaveis.py - F:/Escola18ano/12anoAPIb/Python/2.1 variaveis.py (3.11.5)*
File Edit Format Run Options Window Help
Score = 300
print (Score)
```



Concatenar{

- Quando utilizamos `variáveis` no meio do `print` devemos `concatenar` utilizando o sinal `+`
- A vantagem é que `não acrescenta espaço em branco`, tal como acontecia com a vírgula
- Porém, `não permite misturar texto com números`
- Que raio de `concatenar` é `este` que `não permite misturas`? Não é esse o `objetivo`?
- Bem... temos de `aceitar` as `regras` da `linguagem`. É assim que a linguagem funciona. Compete-nos a nós `dar a volta à situação`

Concatenar{

1 Lembram-se do problema da vírgula?

```
2  
3  
4 *2.2 variaveisguiiao.py - F:/Escola18ano/12anoAPIb/Python/2.2 variaveisguiiao.py (3.11.5)*  
5 File Edit Format Run Options Window Help  
6 print("Eu tenho",18, ".")  
7
```



8 E se utilizarmos o + em vez de ,

```
9 *2.2 variaveisguiiao.py - F:/Escola18ano/12anoAPIb/Python/2.2 variaveisguiiao.py (3.11.5)*  
10 File Edit Format Run Options Window Help  
11 print("Eu tenho"+18+ ".")  
12
```

13 Resulta em...

14 ERROOO!!!! A sério???

```
print("Eu tenho"+18+ ".")  
TypeError: can only concatenate str (not "int") to str
```

Concatenar{

1  Pois... lembram-se de vos dizer que **não é possível misturar texto**
2
3 como **números**?

4
5  Como dar a volta.

6 1 - **Definir** o **número** como **texto** (e **cálculos???**)

7 2 - **Converter** em **texto**

```
9    2.2 variaveisguiiao.py - F:/Escola18ano/12anoAPIb/Python/2.2 variaveisguiiao.py (3.11.5)
10   File Edit Format Run Options Window Help
11   idade="18"
12   print("Eu tenho"+idade+ ".")
```

13 **Eu tenho18.**



```
9    *2.2 variaveisguiiao.py - F:/Escola18ano/12anoAPIb/Python/2.2 variaveisguiiao.py (3.11.5)
10   File Edit Format Run Options Window Help
11   idade=18
12   print("Eu tenho "+str(idade)+ ".")
```

13 **Eu tenho 18.**



Concatenar{

Então e para quê o + afinal???

Porque a vírgula só funciona para o print, caso contrário temos de utilizar o + (e também porque não acrescenta espaços!)

Exemplos:

```
*2.2 variaveisguiiao.py - F:/Escola18ano/12anoAPIb/Python/2.2 variaveisguiiao.py (3.11.5)*
File Edit Format Run Options Window Help
idade= 18
identidade="Eu sou o Ricardo e tenho "+str(idade)+" anos."
print(identidade)
Eu sou o Ricardo e tenho 18 anos.
```

```
2.2 variaveisguiiao.py - F:/Escola18ano/12anoAPIb/Python/2.2 v
File Edit Format Run Options Window Help
texto1="Primeiro texto."
texto2="Segundo texto."
total = texto1+" "+texto2
print(total)
Primeiro texto. Segundo texto.
```

Constantes{

1
2
3  Ao contrário das **variáveis** as **constantes** nunca mudam ao longo do
4 **programa**

5
6  Então **para quê defini-las?**

7  **Simples.** Se forem **atribuídos n^{os}** em vez de constantes num programa
8 e essa constante mudar pontualmente (por exemplo **IVA**) temos de
9 **mudar** em **todos** os **locais** em que a **definimos numericamente**

10  Se **utilizarmos constantes** em vez de **n^{os}** é só **mudar** o seu **valor** que
11 **muda no programa todo**

12  Exemplos:

13 `PI = 3.141593`

14 `IVA = 0.23`

Atenção: o PYTHON não possui uma forma de declarar constantes, pelo que se deve sempre escrever as mesmas em letras **maiúsculas** para se distinguirem das **variáveis**

1
2
3
4
5
6
7
8
9
10
11
12
13
14

Para Hoje{

}
}

Realiza os guiões práticos existentes
no Website da tua disciplina

CREDITS: This presentation template was created by **Slidesgo**,
including icons by **Flaticon**, and infographics & images by **Freepik** and
illustrations by **Stories**

