transforme-se

serasa experian

WISEHANDS
Tecnologia e Acessibilidade

No plano de hoje:

- Variáveis e Constantes
- Operadores Aritméticos
- Strings
- Entradas do Usuário
- IF, ELIF, WHILE, FOR

Variáveis e tipos embutidos

Vamos conhecer os tipos da biblioteca padrão do Python

Os principais tipos internos são **números, sequências, mapas, classes, objetos e exceções**, mas iremos focar primeiramente nos **números e sequências de texto**.

Tipos **embutidos, ou built-ins**, são recursos nativos da linguagem, recursos que já vem prontos para uso.

Números e Textos

Um valor, como um número ou texto, é algo comum em um programa

Por exemplo:

➔ ‘Hello, World!’, 1, 2

São todos valores, mas de diferentes tipos:

➔ 1 e 2 são números inteiros
➔ ‘Hello, World!’ é um texto

Texto também é chamado de String, podemos identificar porque são delimitados por *aspas, simples ou duplas*.

Strings são do tipo str e inteiros do tipo int

Função type()

Caso não tenha certeza qual é o tipo de um valor, pode usar a função type()

Qual é o tipo?

- ➔ 2
- ➔ ‘Hello World’
- ➔ 3.2
- ➔ ‘2’
- ➔ ‘3.2’
- ➔ 2 + 3j

Função type()

Caso não tenha certeza qual é o tipo de um valor, pode usar a função type()

Qual é o tipo?

- ➔ type(2) tipo ‘int’
- ➔ type(‘Hello World’) tipo ‘str’
- ➔ type(3.2) tipo ‘float’
- ➔ type(‘2’) tipo ‘str’
- ➔ type(‘3.2’) tipo ‘str’
- ➔ type(2 + 3j) tipo ‘complex’

Variáveis

Vamos pedir para o Python lembrar de um valor!

O python guarda o valor em uma variável.

O comando de atribuição (sinal de igual =) cria uma variável e atribui o valor.

```
mensagem = ‘oi, mundo’
numero = 5
pi = 3.14
```

Para recuperar esses valores, basta chamar o nome das variáveis.

Nomes ilegais

Não pode começar com número, ter caractere especial ou ser uma palavra reservada

Python possui 33 palavras reservadas:

and	as	assert	break	class	continue	def	del	elif
else	except	FALSE	finally	for	from	global	if	import
in	is	lambda	None	nonlocal	not	or	pass	raise
return	TRUE	try	while	with	yield			

Não podemos usar esses termos para nomear variáveis.

Instruções

É uma unidade de código para o Python executar

Instrução e comando são sinônimos.

Quando executamos um comando no modo interativo, o python apresenta o resultado do comando. Um script é uma sequência de instruções e os resultados vão aparecendo durante a execução do programa.

```
>>> print('Olá mundo!')
```

```
print(1)
x = 2
print(x)
```

Operadores Aritméticos

É uma unidade de código para o Python executar

Operadores são símbolos especiais que representam cálculos.

Uma expressão é uma combinação de valores, variáveis e operadores

Utilizamos os operadores **+, -, *, /** e **** que representam:

adição, subtração, multiplicação, divisão e potenciação.

Também existe o operador **//,** que representa **divisão inteira**

e o **%** que resulta o **resto da divisão de dois números inteiros**

Operadores Aritméticos

Os principais operadores são:

Operação	Nome	Descrição
a + b	adição	Soma entre a e b
a - b	subtração	Diferença entre a e b
a * b	multiplicação	Produto entre a e b
a / b	divisão	Divisão entre a e b
a // b	divisão inteira	Divisão inteira entre a e b
a % b	módulo	Resto da divisão entre a e b
a ** b	exponenciação	a elevado a potência de b

Strings e Operadores

Alguns operadores funcionam com strings

O operador + concatena strings.

texto1 + texto2

O operador * multiplica seu conteúdo por um inteiro

texto1 * 3

Strings também possuem métodos, como .upper() ou .capitalize(). Isso pode ser checado na documentação.

Entrada do Usuário

Vamos criar mais interatividade

A função input() captura a entrada de valores:

- Quando a função é chamada, o programa para e espera o usuário digitar alguma coisa.
- Quando o usuário aperta enter, o programa processa o valor digitado em forma de string.
- Para pedir algo específico ao usuário, podemos passar uma string para a função input().

Convertendo uma string para inteiro

A função input() lê o valor digitado pelo usuário como uma string.

O int também funciona como uma função, que recebe uma string e retorna o inteiro correspondente.

```
numero = int(numero_em_texto)
```

A função format()

A função format() substitui o {} pela variável indicada:

```
nome = input(‘Digite seu nome ‘)
idade = input(‘Digite sua idade ‘)
print(‘Seu nome é {} e sua idade é {}’.format(nome, idade))
```

Constantes

O python possui poucas contantes embutidas.

As mais utilizadas são: **True, False e None.**

Essas também são palavras chaves do Python, portanto, **reservadas e não podem ser usadas como nomes de variáveis.**

Valores booleanos

True e **False** são valores booleanos que representam verdadeiro e falso.

O Python também possui a função bool(), que retorna **True** quando o argumento é verdadeiro e **False** caso contrário.

```
>>> bool(3 > 5)
>>> bool(1 == 1)
>>> bool(0)
>>> bool(‘’)
```

```
>>> bool(None)
>>> bool(1)
>>> bool(-100)
>>> bool(True)
```

Operadores de Comparação

Operação	Descrição
a == b	a igual a b
a != b	a diferente b
a < b	a menor do que b
a > b	a maior do que b
a <= b	a menor ou igual a b
a >= b	a maior ou igual
a is b	True es a e b são idênticos
a is not b	True se a e b não são idênticos
a in b	True se a é membro de b
a is not in b	True se a não é membro de b

Comando if

Operador condicional

```
idade = 18
if idade >= 18:
    print('maior de idade')
else:
    print('menor de idade')
```

Comando elif

Operador condicional

```
idade = 18
if idade < 12:
    print('crianca')
elif idade < 18:
    print('adolescente')
elif idade < 60:
    print('adulto')
else:
    print('idoso')
```

Exercício:

Jogo da adivinhação

Crie um algoritmo, em português estruturado, que, dado um número secreto, leia um valor de entrada feito pelo usuário e verifique se o valor é igual, maior ou menor do que o número secreto.

Lembrando, as principais características de um algoritmo são: finitas, bem definidas e efetivas.

Comando while

Operador de laço

```
x = 5
while(x > 1):
    print(x)
    x = x - 1
```

Exercício:

Modificando o Programa

Complemente o algoritmo criado anteriormente:

- Caso haja necessidade, corrija de acordo com o que foi discutido em sala
- Acrescente, ainda em portugês estruturado, um limite de tentativas que o usuário tem para acertar o número secreto. O algoritmo deverá informar a quantidade de rodadas que faltam, bem como em que rodada está no momento.

Função range()

Define uma série de valores

O range(1, 10) vai gerar os números de 1 a 9.

Para definir um passo, um intervalo entre os elementos, passamos um terceiro valor, range(1, 10, 2).

Por padrão, o passo sempre é 1.

Comando for

Operador de laço

Usando a função range():

```
for x in range(1,10):
    print(x)
```

Podemos passar os valores da sequência manualmente:

```
for x in [1,2,3,4,5,6]:
    print(x)
```

Exercício:

Utilizando o for no jogo e novas funcionalidades

- Substitua o comando while pelo for
- Escolha um número secreto entre 0 e 100.
- Crie um nível de dificuldade para o jogo. Peça para o usuário escolher em qual nível ele deseja jogar. O nível é mensurado de acordo com as tentativas.
- Acrescente uma sistema de pontuação. O jogador deve começar com 1000 pontos e cada chute errado deve ser subtraído do total de pontos um valor que corresponde a diferença entre o chute e o número secreto. (dica, pesquise sobre a função abs(), ela pode ser útil)

transforme-se

O conhecimento é o poder
de transformar o seu futuro.