IEC081
Introdução à Programação de Computadores
Aula 08 – Variáveis

Turma: Física

Professor: André Luiz da Costa Carvalho

E-mail:andre@icomp.ufam.edu.br

Página:iccufam.wordpress.com

Conteúdo

- Variáveis
- Entrada e saída de dados
- Tipos de variáveis
- Boas práticas de programação
- Erros

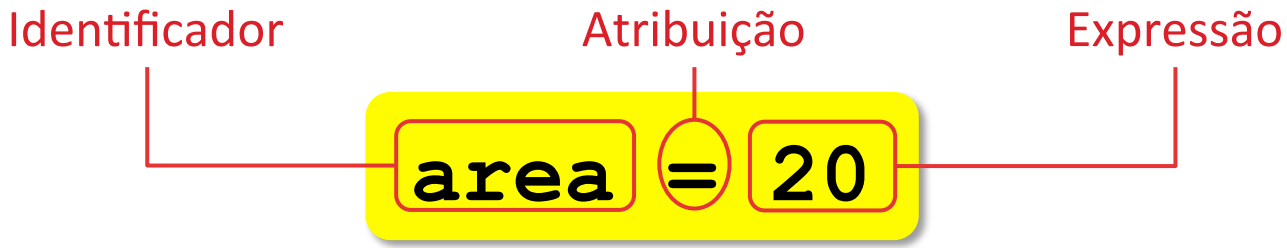
Conteúdo

- Variáveis
- Entrada e saída de dados
- Tipos de variáveis
- Boas práticas de programação
- Erros

O que são variáveis?

- Variável é uma região de memória do computador que contém um valor, conhecida por um nome especificado pelo usuário.

Como definir variáveis?



Identificador

- Nome dado aos objetos utilizados no programa (variáveis, constantes, funções, etc.)

Atribuição

- Comando que define ou redefine o valor armazenado em uma variável

Expressão

- Pode ser um valor ou um conjunto de comandos que resulta em um valor

Regras para identificar variáveis

- O nome da variável deve iniciar obrigatoriamente com uma letra do alfabeto ou o caractere sublinhado (_).
- Os demais caracteres podem conter letras, números ou o caractere sublinhado (_).
- Não use uma palavra-chave reservada:

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

Regras para identificar variáveis
:: Cuidados

- Não use espaços.
- Letras maiúsculas e minúsculas são diferentes.
  - Variáveis **`Area`** e **`area`** são distintas.

Regras para identificar variáveis
:: Exemplos

Nome	Válido	Comentário
`dia1`	✓	
`diaDaSemana`	✓	
`dia da semana`	✗	Contém espaços
`dia_da_semana`	✓	
`dia#3`	✗	Usa símbolo inválido
`3o_dia`	✗	Começa com número
`_dia`	✓	

Variáveis
:: Observações

- Se você declarar uma variável já existente, o conteúdo anterior será perdido.

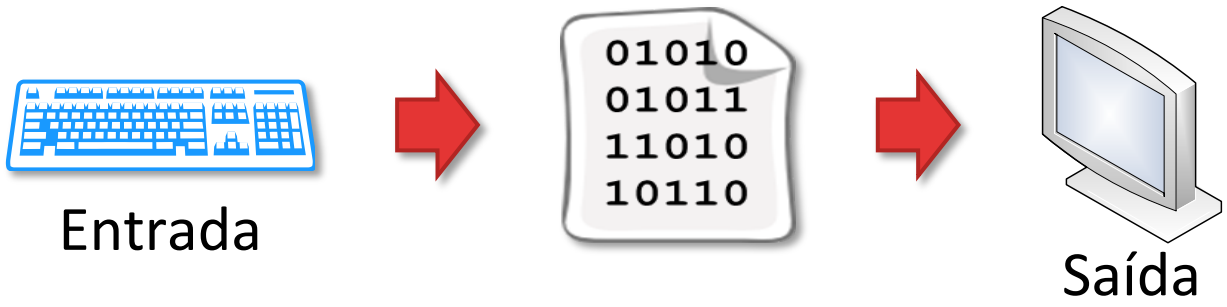
```
>>> a = 6
>>> a
6
>>> a = 22
>>> a
22
```

Conteúdo

- Variáveis
- **Entrada e saída de dados**
- Tipos de variáveis
- Boas práticas de programação
- Erros

Entrada e Saída de Dados
:: Operação Básica

- Entrada são os dados que o programa recebe durante sua execução.
- Uma forma comum de entrada é pelo teclado.
- Uma vez que a entrada é recebida ela é normalmente processado pelo programa.
- Os resultados do processamento são, então, exibidos na tela como saída programa.



Programação
:: Papéis

Usuário

- Utiliza o script
- Insere dados

Programador

- Projeta e escreve o script

Comandos de Entrada e Saída

Comandos de Entrada

- Permitem que o usuário do programa entre com novos valores sem alterar o código do script.
- Exemplo:
  - Função `input()`

Comandos de Saída

- Exibem resultados no Shell.
- Exemplos:
  - Função `print()`

Saída de dados

:: print()

- Exibe os dados armazenados na variável usada como argumento.

```
>>> print("Oi")
```

```
>>> x = 109
>>> print(x)
```

```
>>> texto = "ola mundo"
>>> print(texto)
```

Entrada de dados pelo usuário

:: `input()`

- Exibe, no Shell, um texto de orientação para o usuário e aguarda que este digite um valor.

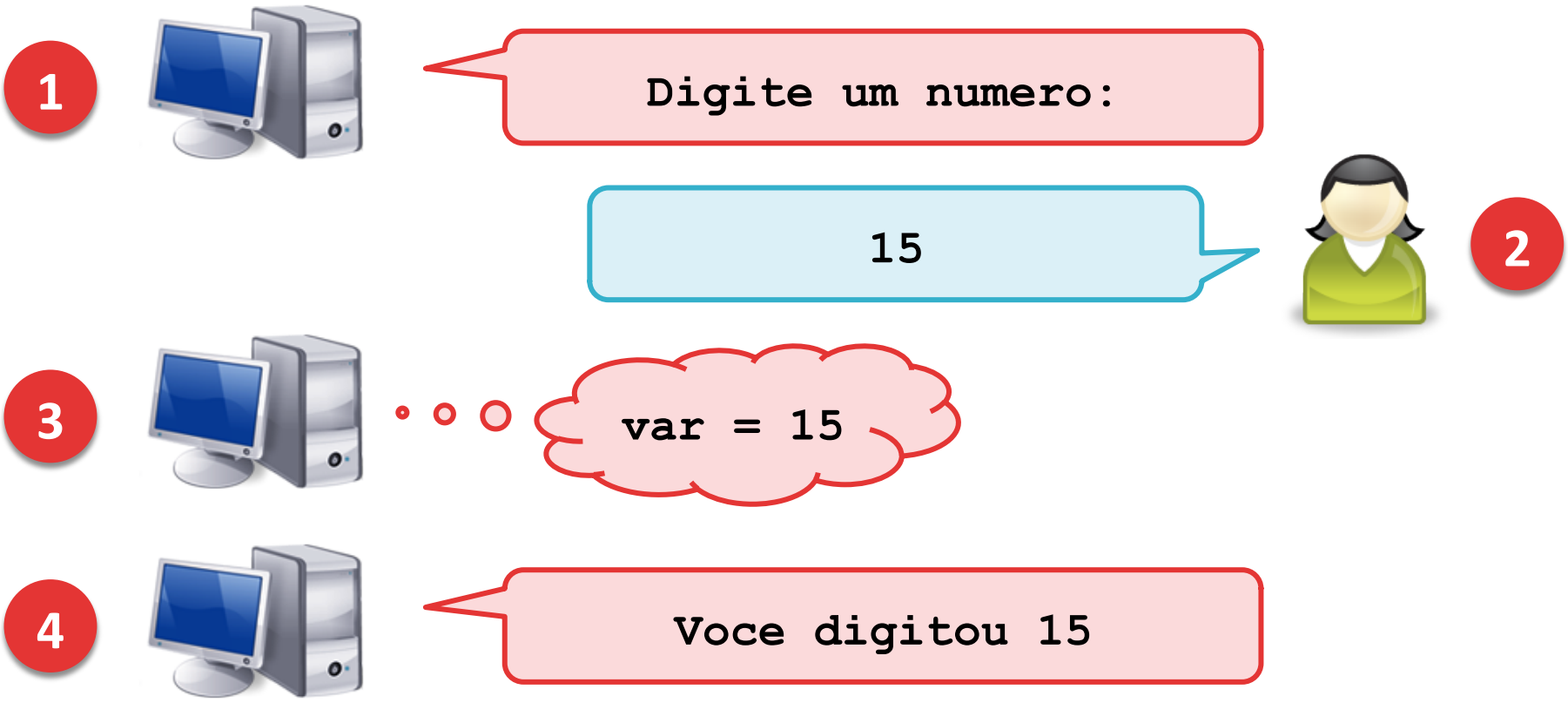
```
>>> var = input("Digite um numero: ")
```

- O Shell ficará travado até que o usuário digite um valor.
- O valor digitado será armazenado na variável **`var`**.

Entrada e Saída de Dados

:: Exemplo

```
>>> var = input("Digite um numero: ")
>>> print("Voce digitou", var)
```



Entrada de Dados

pelo Usuário

- Dinâmico: o script pode ser usado para diversas instâncias do problema

```
cat1 = input("Digite
o valor do cateto: ")
```

pelo Script

- Estático: o script tem de ser modificado para cada instância do problema

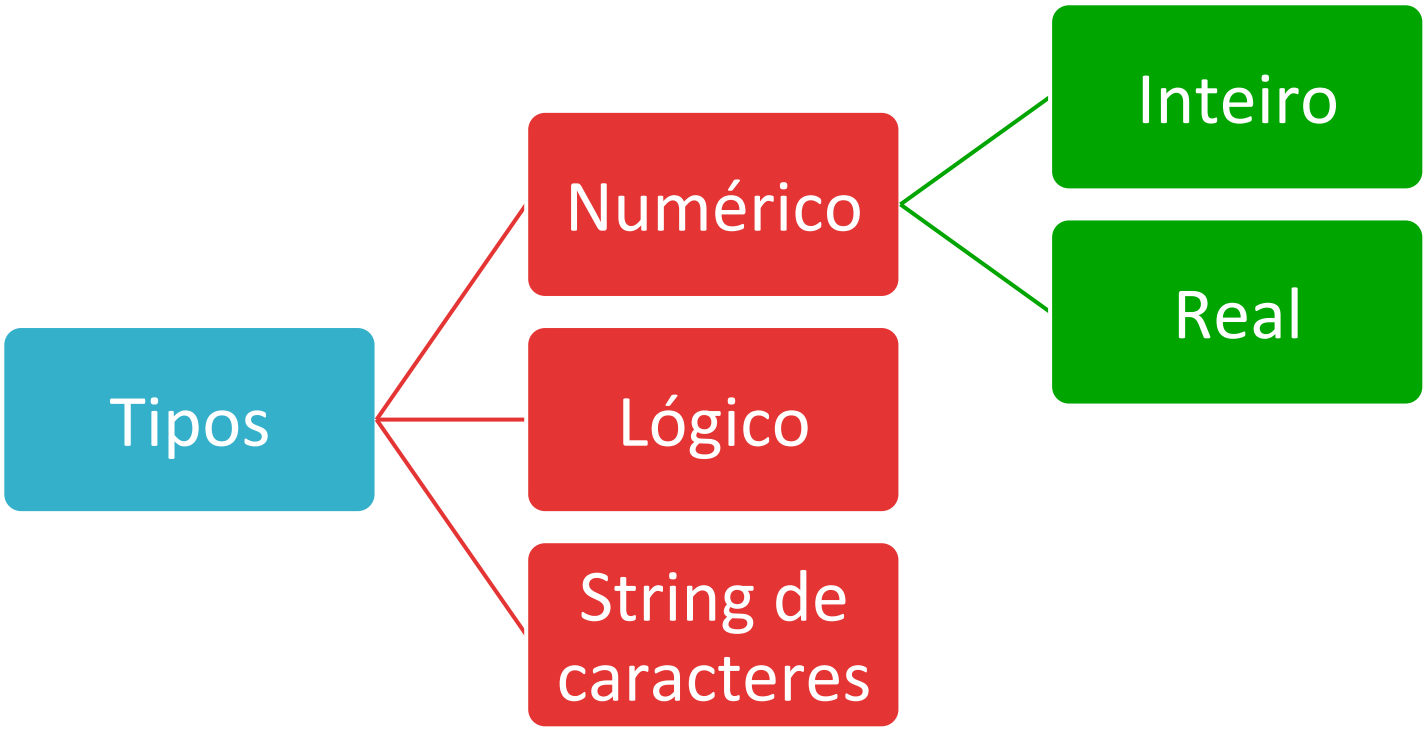
```
cat1 = 3
```

Conteúdo

- Variáveis
- Entrada e saída de dados
- **Tipos de variáveis**
- Boas práticas de programação
- Erros

Tipos de dados

- O Tipo define a natureza dos dados que a variável armazena.
- Tipos mais comuns no Python:



Tipos Numéricos

:: Classificação

Inteiros (`int`)

- São números sem a parte fracionária.
- Exemplos: 1 | 0 | -5 | 567

Reais (`float`)

- São números com parte fracionária.
- Também conhecidos como ponto flutuante.
- Exemplos: 1.0 | 3.1415 | 2.7182

Tipos Numéricos
:: Observações

- Números inteiros e de ponto flutuante são representados de maneiras distintas na memória do computador.
- Por isso, $1 \neq 1.0$
- Em Python e na maioria das linguagens de programação, utilizamos o ponto – e não a vírgula – como separador entre a parte inteira e a parte fracionária de um número ponto flutuante.

Tipos Numéricos

:: Exemplos

Número	Inteiro	Ponto flutuante
`5`	✓	
`5.0`		✓
`4.3`		✓
`-2`	✓	
`100`	✓	
`1.333`		✓

Representação de números reais :: Observações (1)

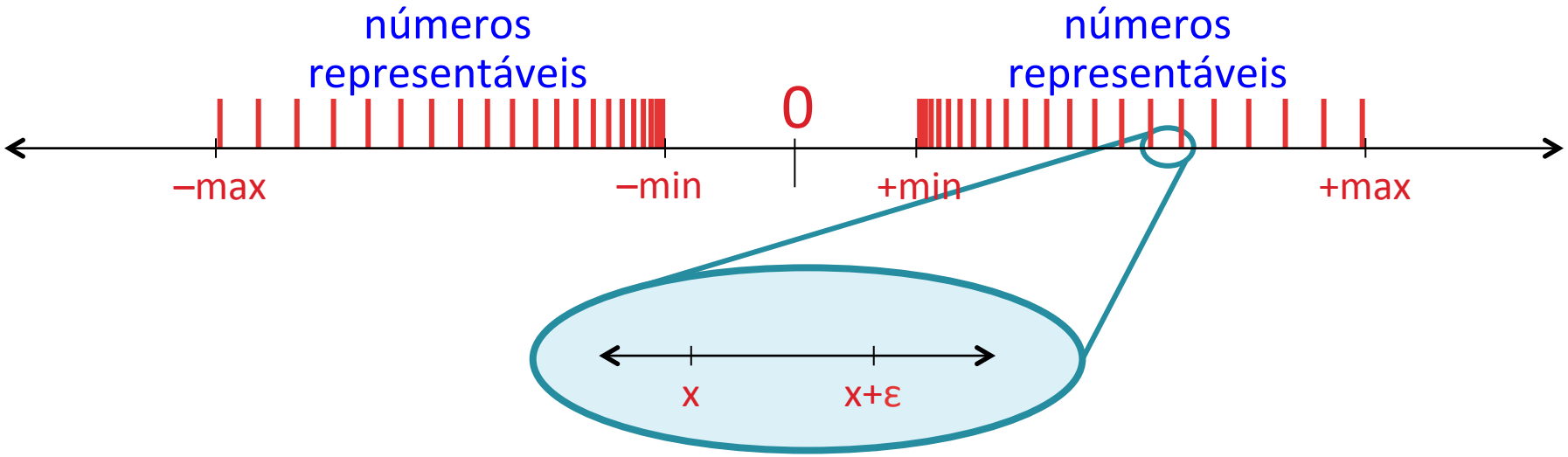
- Variáveis de ponto flutuante são representações da realidade na memória do computador.
- O conjunto dos números reais é infinito, mas o espaço de armazenamento em memória é um recurso finito.
- Logo, somente alguns elementos do conjunto de números reais podem ser representados em um computador.

Representação de números reais :: Observações (2)

- Variáveis de ponto flutuante são guardadas no formato exponencial:

$$\text{sinal} \times \text{mantissa} \times \text{base}^{\text{expoente}}$$

- Épsilon (ε): pequeno intervalo entre cada número real representável em computador e seu vizinho mais próximo.
- O valor de ε varia conforme a grandeza representada.



Representação de números reais
:: Exemplo

```
>>> 10**3  + 0.0001
1000.0001
```

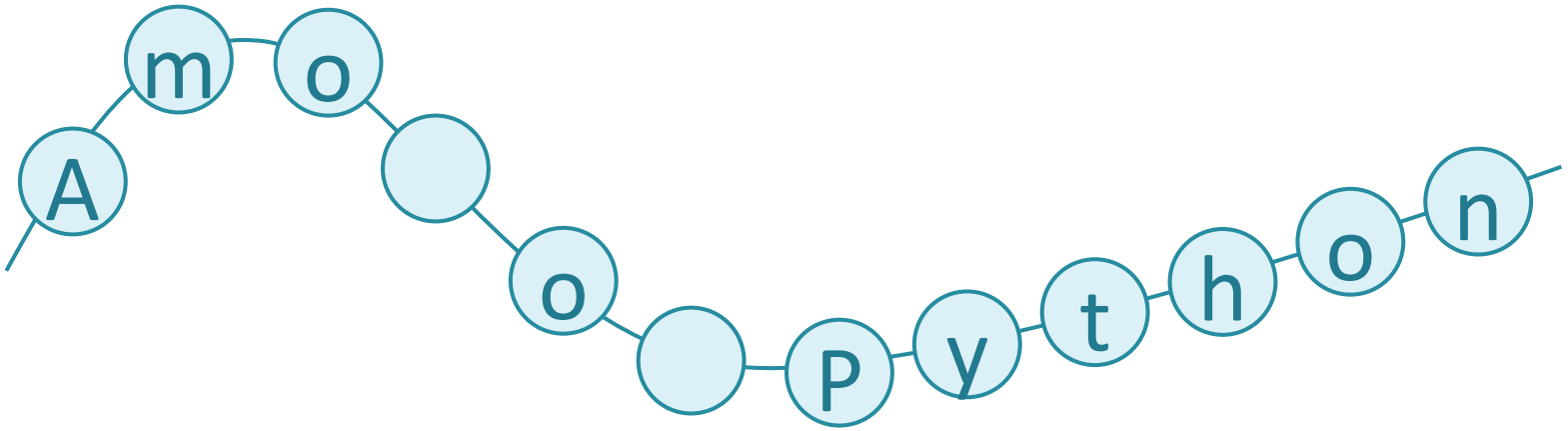
```
>>> 10**30 + 0.0001
1e+30
```

Tipo Lógico

- Uma variável do tipo lógico (ou booleano) armazena um conteúdo que assume apenas um de dois valores possíveis:
  - `True` (verdadeiro)
  - `False` (falso)
- Note que as iniciais T e F são escritas em maiúsculas.

Strings de caracteres

- Uma string (= corda) é uma cadeia de caracteres.
- Uma cadeia de caracteres é um sequência de símbolos, tais como letras, números, sinais de pontuação, etc., que formam textos em geral.



Strings de caracteres

- O início e o fim de uma string são indicados por aspas ("), de modo a separar o conteúdo da string do restante do texto do programa.

```
texto = "Amo o Python"
```

- Note que você pode usar espaços dentro de uma string de caracteres.

Strings de caracteres

- Se você quiser incluir o símbolo de aspas em uma string, use a expressão \".

```
>>> texto2 = "Amo o \"Python\"."
>>> print(texto2)
Amo o "Python".
```

Tipos e Input

- O input() por padrão, no Python 3.x, retorna uma string.
  - Não posso fazer contas com strings
- Como proceder para ter um int através de input()?
  - Cast
  - Variavel = int(input(“Digite um numero”)
  - Var_real = float(input(“Digite um numero real”)

Conteúdo

- Variáveis
- Entrada e saída de dados
- Tipos de variáveis
- **Boas práticas de programação**
- Erros

Boas práticas de programação

:: Comentários

- Códigos devem ser escritos para serem lidos por seres humanos.
- Escreva os comentários no momento em que estiver escrevendo o código.
- Os comentários devem acrescentar informação, e não frasear o comando:

```
# Multiplicacao de b por h:
area = b * h
```

```
# Calculo da area do retangulo:
area = b * h
```

Boas práticas de programação

:: Comentários

- Faça um cabeçalho início do arquivo para explicar a finalidade do script

```
#----------------------------------------
# UNIVERSIDADE FEDERAL DO AMAZONAS
# FULANO DA SILVA
# DATA: 25/04/2013
# ULTIMA MODIFICACAO: 26/04/2013
#
# OBJETIVO: Calcular o volume de combustivel
#           em um tanque cilindrico
#----------------------------------------
```

Boas práticas de programação
:: Identificadores

- Sempre use nomes descritivos e fáceis de lembrar para suas variáveis:

```
x = 1.3
```

```
raio = 2.2
```

- Use sempre letras minúsculas em nomes de variáveis:

```
raio = 1.3
Raio = 4.6
RAIO = 7.9
```

```
raio_interno = 1.3
raio_meio    = 4.6
raio_externo = 7.9
```

Boas práticas de programação

- Defina todas as variáveis que você vai utilizar no início de cada script, a fim de tornar mais fácil a manutenção do código.

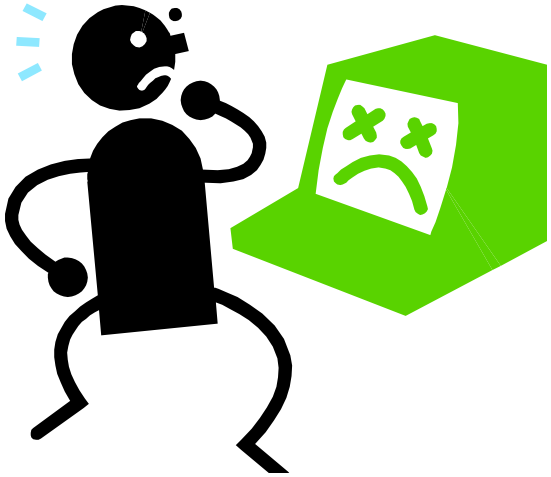
```
nivel  = 0.8        # nivel de combustivel (m)
altura = 2.3        # altura do tanque (m)
raio   = 1.5        # raio da secao vertical (m)
volume = 0          # volume de combustivel (m3)
```

Conteúdo

- Variáveis
- Entrada e saída de dados
- Tipos de variáveis
- Boas práticas de programação
- Erros

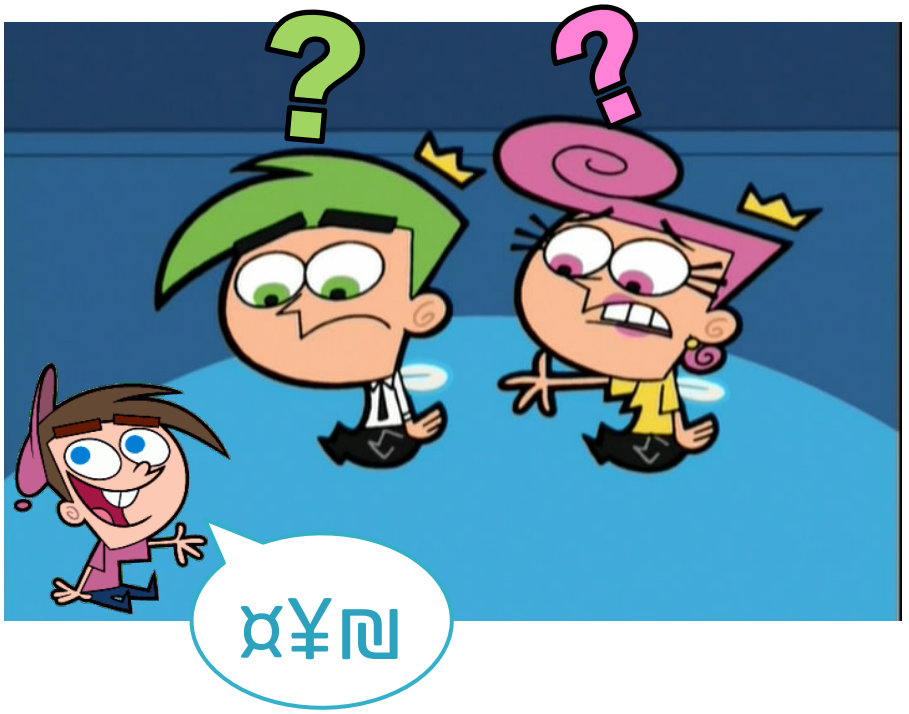
Erros

- Lidar com erros faz parte do desenvolvimento de programas.
- Erros de sintaxe
- Erros de execução

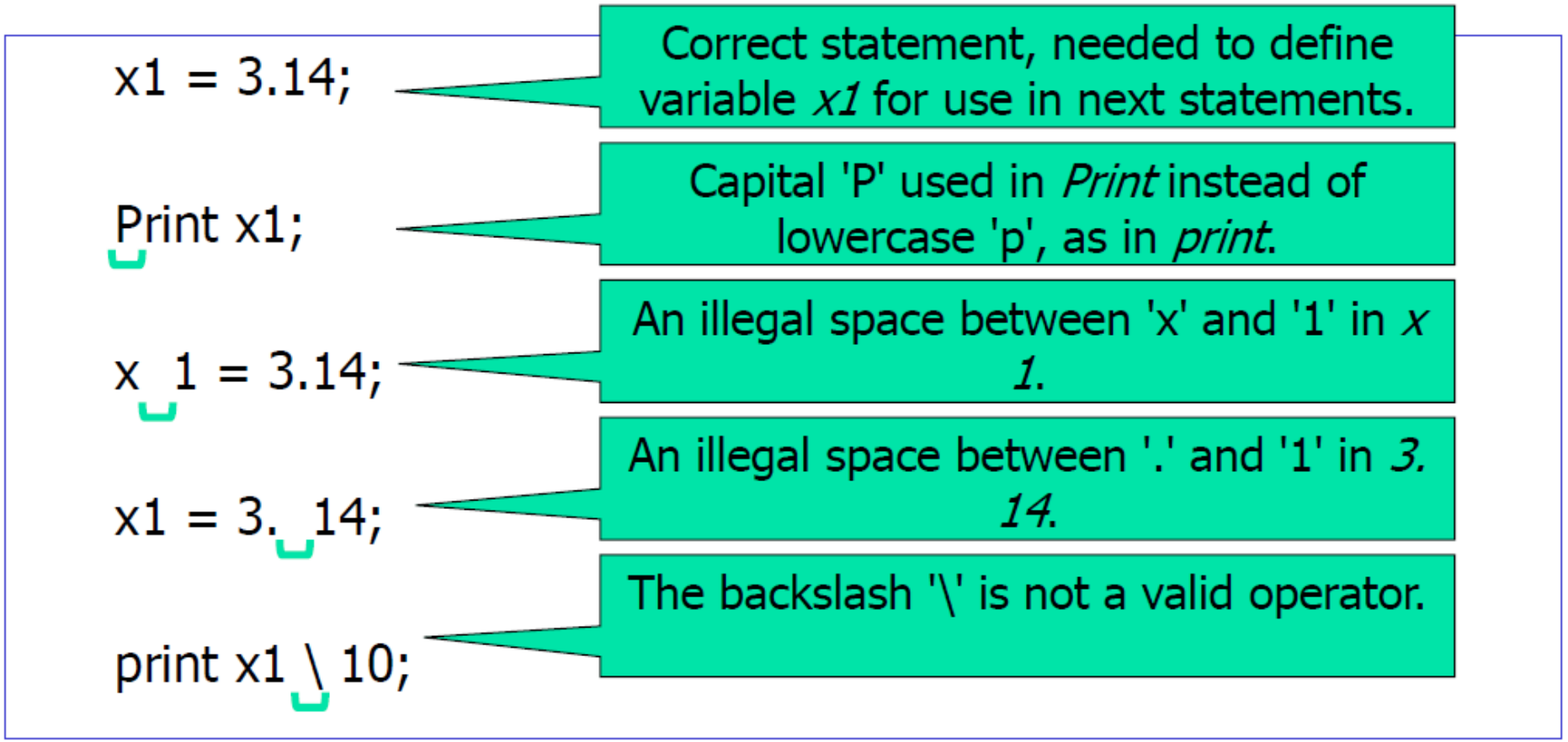


Erros de sintaxe

- Os erros de sintaxe são violações das regras de escrita da linguagem.
- Quando um programa Python contém um erro de sintaxe, uma mensagem de erro de sintaxe é produzida pelo interpretador.
- Programas com erros de sintaxe não são executados.



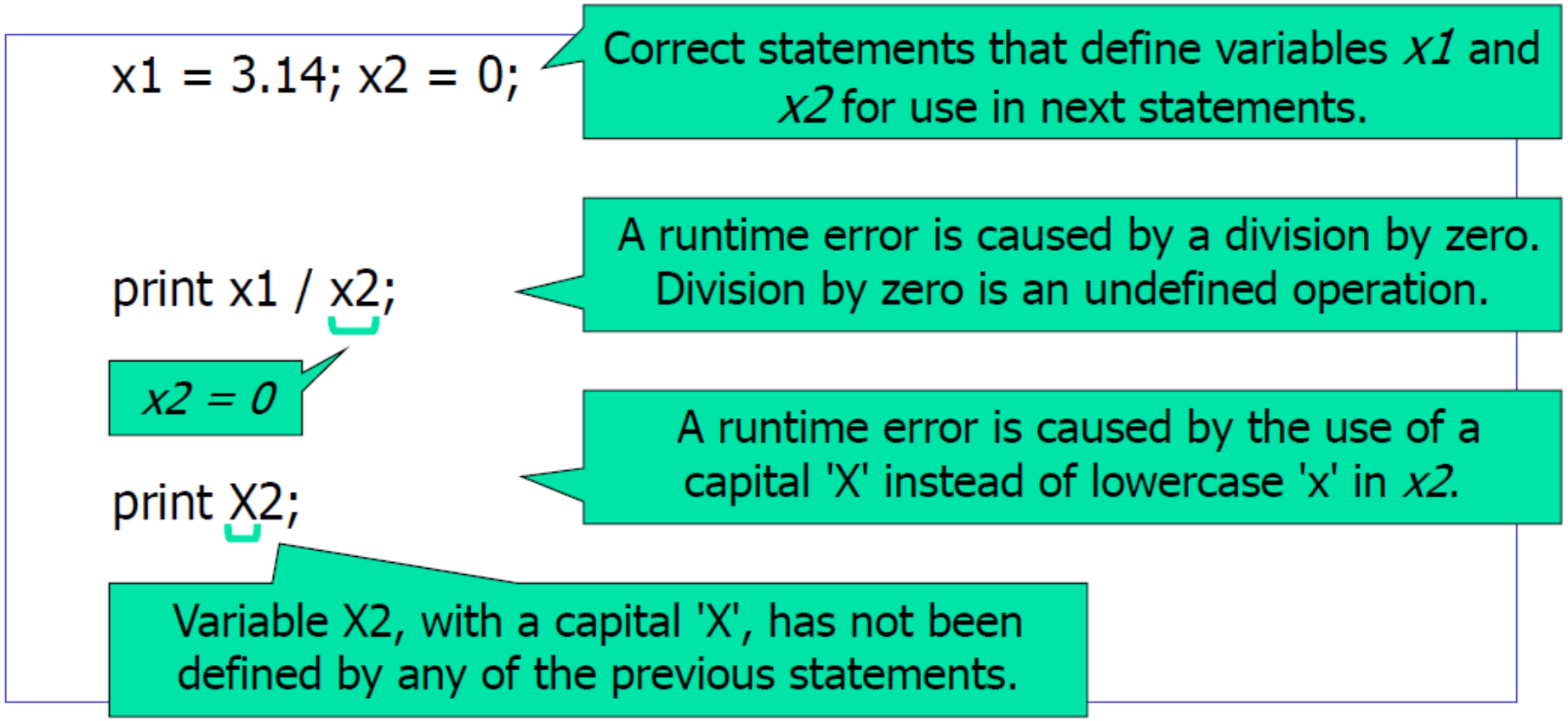
Erros de sintaxe



Erros de execução

- Erros de execução são violações da semântica da linguagem.
- A maioria dos erros de tempo de execução não pode ser detectada antes da execução do programa.

Erros de execução



Referências bibliográficas

- Menezes, Nilo Ney Coutinho (2010). Introdução à Programação com Python. Editora Novatec.
- Farrer, Harry (2011). Algoritmos Estruturados, 3ª edição. Editora LTC.
- Gaddis, Tony (2012). Starting out with Python, 2ª edição. Editora Addison-Wesley.

Dúvidas?