

CodeCity

aprenda Python construindo sua
metrópole sustentável





VARIÁVEIS



Guardando recursos da cidade

VARIÁVEIS



Guardando recursos da cidade

No mundo de **CodeCity**, cada recurso que você coleta precisa de um nome para ser lembrado. Em Python, usamos variáveis para guardar informações.

```
● ● ● CodeCity - Geiza Morais.py  
1 # Guardando quantidade de energia solar coletada  
2 energia_solar = 50  
3  
4 # Guardando número de bicicletas disponíveis  
5 bicicletas = 12  
6  
7 print("Energia Solar:", energia_solar)  
8 print("Bicicletas:", bicicletas)
```

- Python cria espaços na memória com nomes (variáveis)
- Guarda informações nesses espaços
- Você pode usar e imprimir essas informações quando quiser

Dica de jogador: Cada variável é como um depósito na cidade – sem nome, você não sabe o que tem!



TIPOS DE DADOS

Diferentes recursos



TIPOS DE DADOS



Diferentes recursos

Em Python, os dados vêm em vários tipos: números, textos, listas... Cada tipo é um recurso diferente que você administra.



CodeCity - Geiza Morais.py

```
1 # Número de árvores plantadas
2 arvores = 120      # inteiro
3
4 # Nome do bairro
5 bairro = "Centro Verde" # texto
6
7 # Lista de edifícios sustentáveis
8 predios = ["Solar Tower", "Wind Plaza", "EcoHub"] # lista
9
10 print(bairro, "tem", arvores, "árvore e prédios como", predios)
```

- Python reconhece o tipo de cada dado: número, texto ou lista
- Cada tipo pode ser usado de forma diferente no código
- Você pode combinar e manipular esses dados para organizar a cidade



Dica de jogador: Conhecer os tipos de dados é como saber se você está lidando com energia, água ou alimentos. Cada um precisa de cuidados diferentes!



ABRAS

CONDICIONAIS

Decidindo o que construir

CONDICIONAIS

Decidindo o que construir



Você quer construir um parque ou uma fábrica? Python usa if para tomar decisões com base nos recursos da cidade.

```
CodeCity - Geiza Morais.py

1 energia_solar = 80
2
3 if energia_solar > 50:
4     print("Podemos abrir mais um parque!")
5 else:
6     print("Precisa coletar mais energia primeiro.")
```

- Python verifica se a condição é verdadeira (`energia_solar > 50`)
- Se for, executa o bloco do `if`
- Se não for, executa o bloco do `else`

Dica de jogador: Condicionais ajudam a cidade a se manter sustentável.



LOOPS

Automatizando tarefas diárias



Automatizando tarefas diárias

No mundo de CodeCity, você precisa fazer tarefas repetitivas, como distribuir bicicletas ou regar plantas. Python tem loops que repetem uma ação sem você ter que escrever o código várias vezes. Exemplo simples: entregar bicicletas aos bairros



CodeCity - Geiza Moraes.py

```
1 bairros = ["Centro Verde", "Jardim Solar", "Praça da Lua"]
2
3 for bairro in bairros:
4     print("Entregando bicicletas para o bairro:", bairro)
```

- Python pega cada bairro da lista, um por um
- Executa a ação (imprime a mensagem)
- Passa para o próximo bairro automaticamente

 Dica de jogador: Loops são como rotas de entrega automáticas da sua cidade. Você programa uma vez e todas as tarefas repetitivas acontecem sozinhas!



FUNCÕES

Habilidades especiais



Habilidades especiais

Funções são como poderes especiais. Você programa uma vez e pode usar várias vezes na cidade.



CodeCity - Geiza Moraes.py

```
1 # Definindo a função
2 def plantar_arvores(qtd):
3     for i in range(qtd):
4         print(f"Árvore {i+1} plantada na cidade!")
5
6 # Usando a função
7 plantar_arvores(3)
8 plantar_arvores(5)
```

- Python guarda o bloco de código com um nome (função)
- Quando você chama a função, o código dentro dela é executado
- Você pode chamar várias vezes com diferentes valores (quantidades)

 Dica de jogador: Quanto mais funções você domina, mais eficiente é seu gerenciamento urbano!



AGRADECIMENTOS

OBRIGADA POR LER ATÉ AQUI!



Este e-book foi criado utilizando IA e diagramado por uma humana - eu mesma.

O passo-a-passo se encontra no meu Github.

Esse conteúdo foi gerado para fins didáticos de aprendizado de como construir um e-book, e não foi realizada uma avaliação cuidadosa sobre seu conteúdo podendo conter erros gerados pela IA.



<https://github.com/m0ra1s/>