



# Revisão Geral 1

fmasanori@gmail.com

# Acabamos o primeiro livro



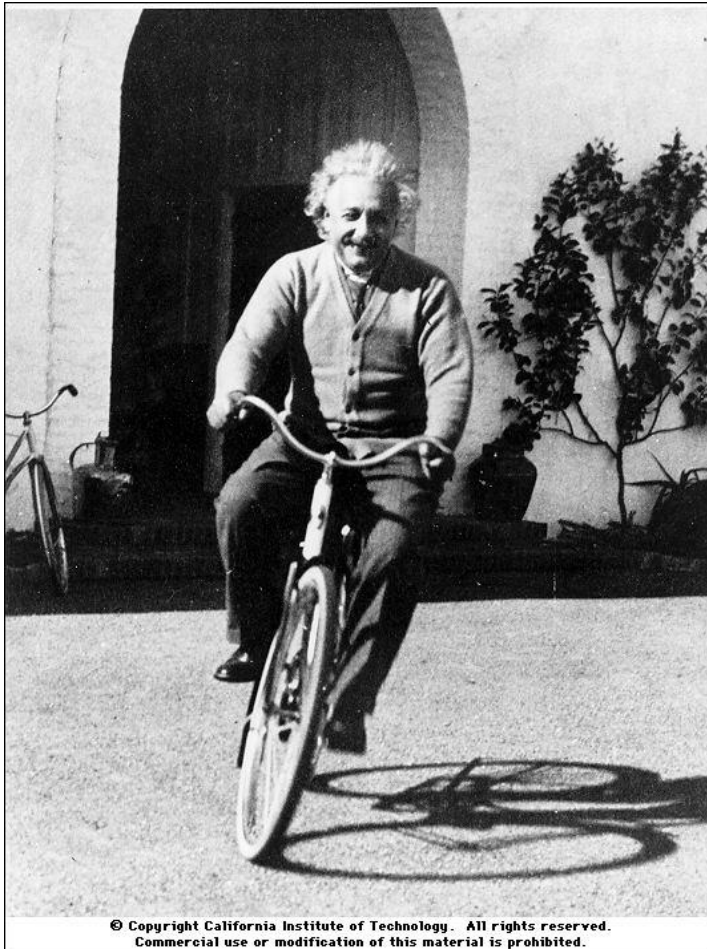
# Vamos começar o segundo



# O que aprendemos

- Variáveis e entrada de dados
- Condições
- Repetições
- Listas
- Strings
- Funções
- Arquivos
- Dicionários
- Classes e Objetos

# Agora só faltam seus exercícios



*“A vida é como  
andar de bicicleta.  
Para manter o  
equilíbrio, é preciso  
se manter em  
movimento”.*  
*Einstein.*

# O que vamos aprender no segundo?

- As mesmas coisas!
- Fazendo jogos
- Acessando sites para ver a cotação do café
- Usando interfaces gráficas
- Tratando exceções
- Usando banco de dados de surfistas
- Mixando músicas

# Encontrando seu caminho

- Se você só usar o software dos outros, sempre estará limitado àquilo que outras pessoas acham que você deseja fazer
- Escreva seus próprios programas
- Você deseja ser programado ou ser o programador?
- Você pode *assumir o controle*

# Adivinhando números

## Funções

```
print ('Bem vindo!')
g = input ('Chute um número: ')
chute = int(g)
if chute == 42:
    print ('Você venceu!')
else:
    print ('Você perdeu!')
print ('Fim do jogo!')
```



# Adivinhando números

Strings

```
print ('Bem vindo!')
g = input ('Chute um número: ')
chute = int(g)
if chute == 42:
    print ('Você venceu!')
else:
    print ('Você perdeu!')
print ('Fim do jogo!')
```

# Adivinhando números

Variáveis

```
print ('Bem vindo!')
g = input ('Chute um número: ')
chute = int(g)
if chute == 42:
    print ('Você venceu!')
else:
    print ('Você perdeu!')
print ('Fim do jogo!')
```

# Adivinhando números

Diretivas

```
print ('Bem vindo!')
g = input ('Chute um número: ')
chute = int(g)
if chute == 42:
    print ('Você venceu!')
else:
    print ('Você perdeu!')
print ('Fim do jogo!')
```

# Adivinhando números

```
print ('Bem vindo!')
g = input ('Chute um número: ')
chute = int(g)
if chute == 42:
    print ('Você venceu!')
else:
    print ('Você perdeu!')
print ('Fim do jogo!')
```

Identação

# Adivinhando números

```
print ('Bem vindo!')
g = input ('Chute um número: ')
chute = int(g)
if chute == 42:
    print ('Você venceu!')
else:
    print ('Você perdeu!')
print ('Fim do jogo!')
```



= atribuição

# Adivinhando números

```
print ('Bem vindo!')
g = input ('Chute um número: ')
chute = int(g)
if chute == 42:
    print ('Você venceu!')
else:
    print ('Você perdeu!')
print ('Fim do jogo!')
```



**== comparação**

# As partes do seu programa

- **Funções** embutidas (*built in functions*) são chamadas pelo seu nome e exigem parênteses
- **Strings** se diferenciam dos comandos do programa por estarem dentro de aspas
- **Variáveis** controlam dados na memória e possuem tipos diferentes
- **Diretivas** são comandos da linguagem

# As partes do seu programa

- A **identação** separa blocos de comandos
  - “Cada um no seu quadrado”
- Um igual (=) significa *atribuição*
  - Ex.: chute = int(g) (chute recebe inteiro de g)
- Dois iguais (==) significa *comparação*
  - Ex.: chute == 42 (chute é igual a 42?)
- Dois pontos abrem blocos de instruções
  - Coloque no banheiro, caderno, geladeira:
  - “Eu amo dois pontos!”



# Quais os tipos de erro?

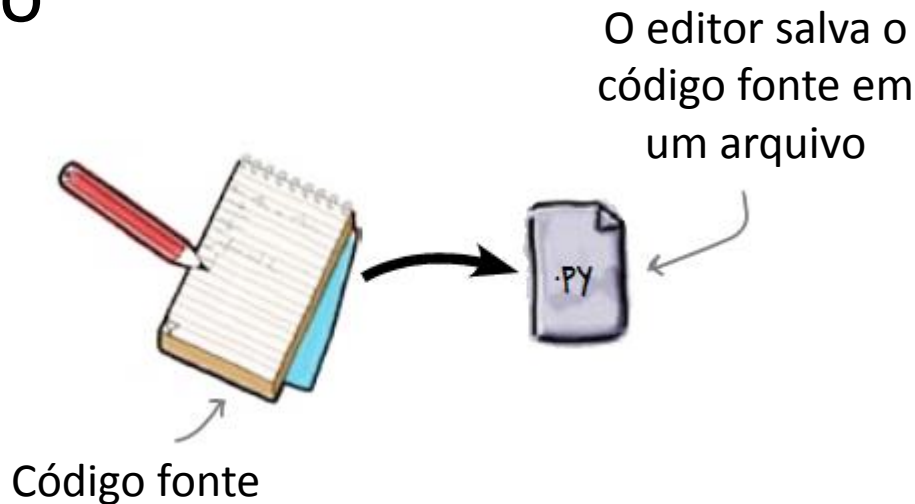
- Erros de sintaxe
  - Uma linguagem de programação é formal, diferente das linguagens naturais possui sintaxe rígida
- Erros em tempo de execução
- Erros semânticos (mais difíceis de achar)

# Como achar e tratar erros?

- Sintáticos: muita atenção e prática
- Em tempo de execução: tratamento de exceções
- Semânticos: teste de mesa ou simulação

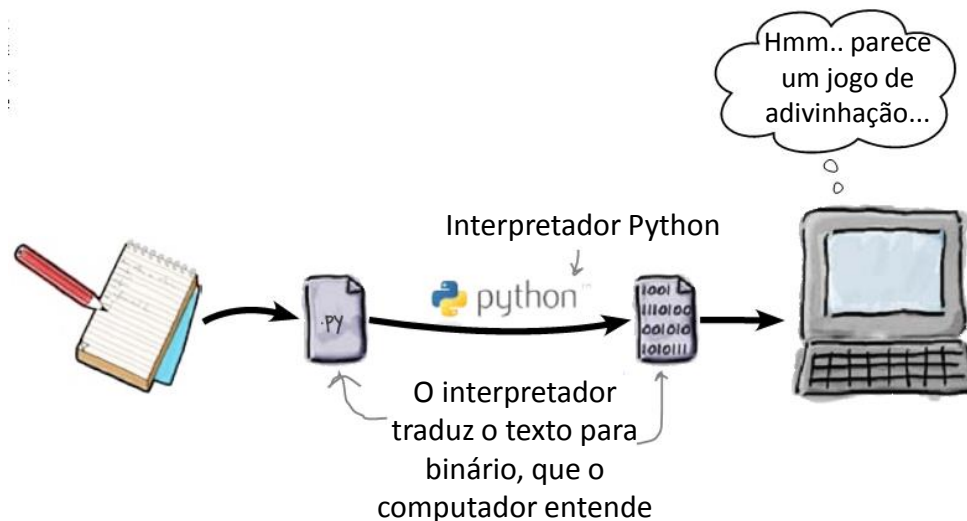
# Então, como você executa seu código?

- Há duas coisas para executar o programa de adivinhação: um *editor* e um *interpretador*
- O editor salva o código escrito em um arquivo no disco



# Então, como você executa seu código?

- Computadores não podem processar textos pois somente entendem binário (zeros e uns)
- O interpretador transforma o código fonte em um arquivo binário para o computador



# Então, como você executa seu código?

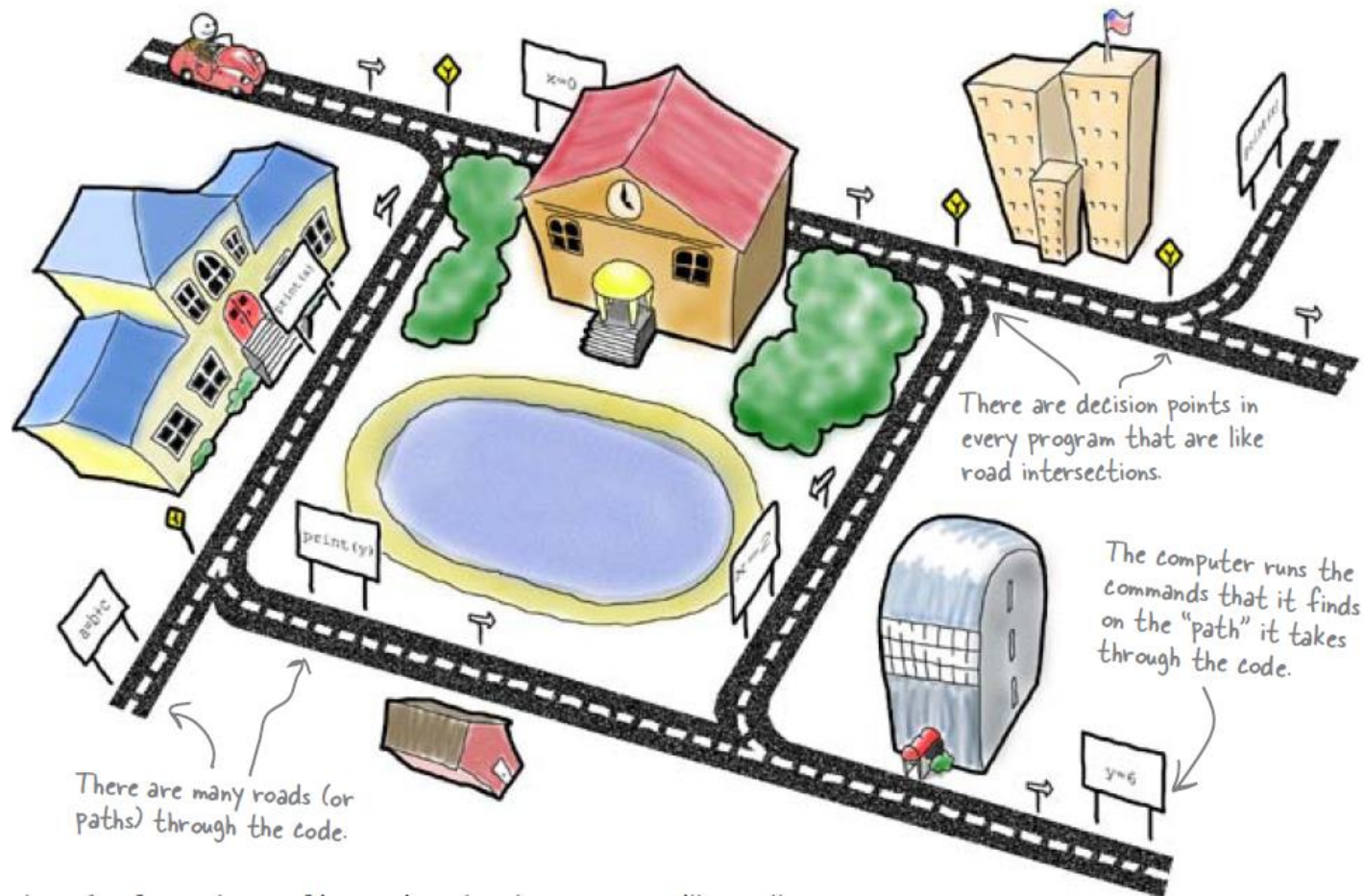
- O interpretador Python atua em dois modos: interativo e edição
- O modo interativo é ótimo para testar comandos e obter respostas instantâneas
- Porém o modo edição é o mais utilizado para desenvolver os programas
  - Nomes dos arquivos geralmente terminam com “.py”
  - Caso utilize outra extensão perderá as cores...

Um programa é mais que uma lista de comandos

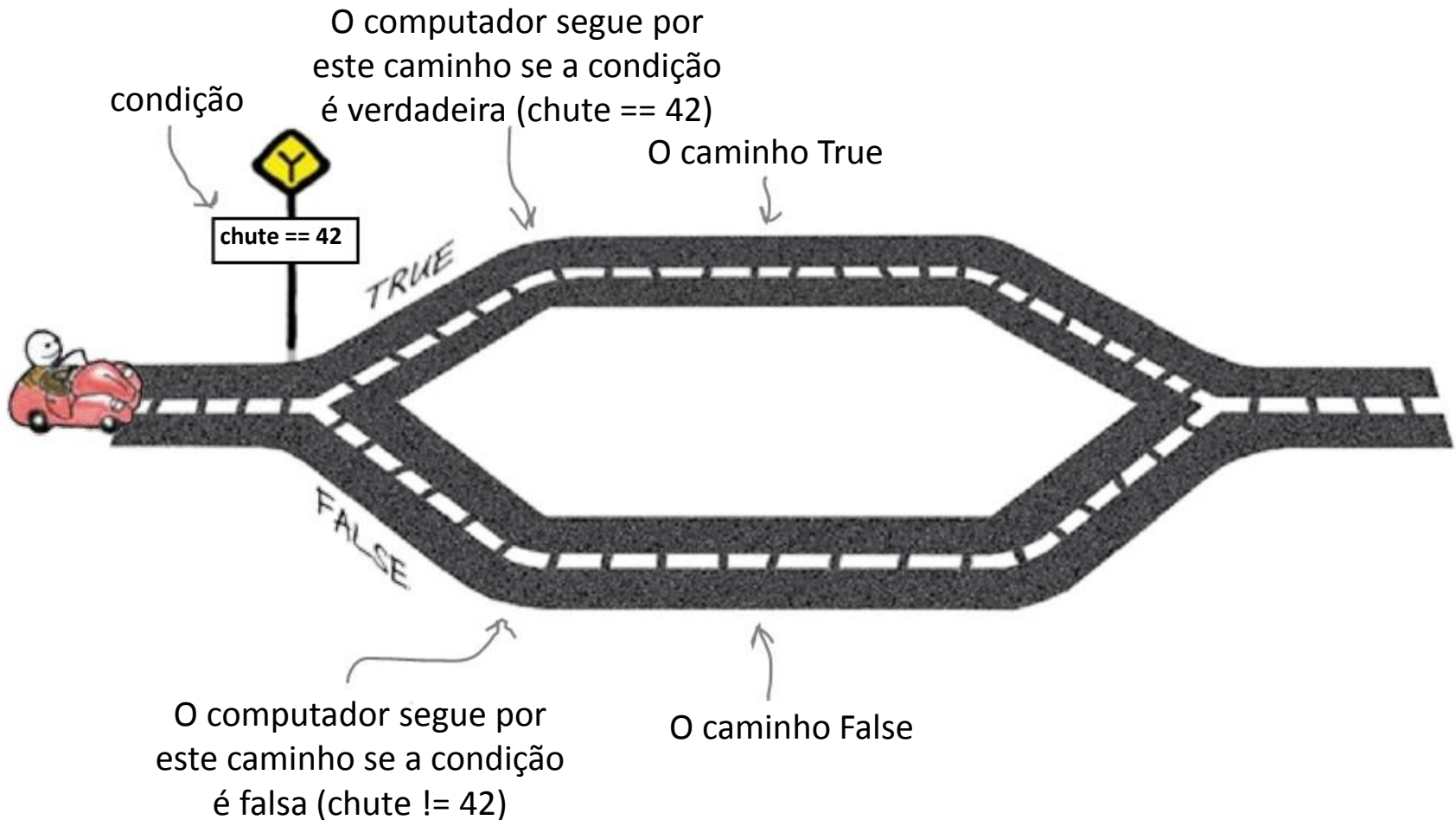
```
print ("Bem vindo ao meu programa!")  
print ("Volte sempre!")
```



# O programa é uma rede viária



# Na rede você escolhe seu caminho

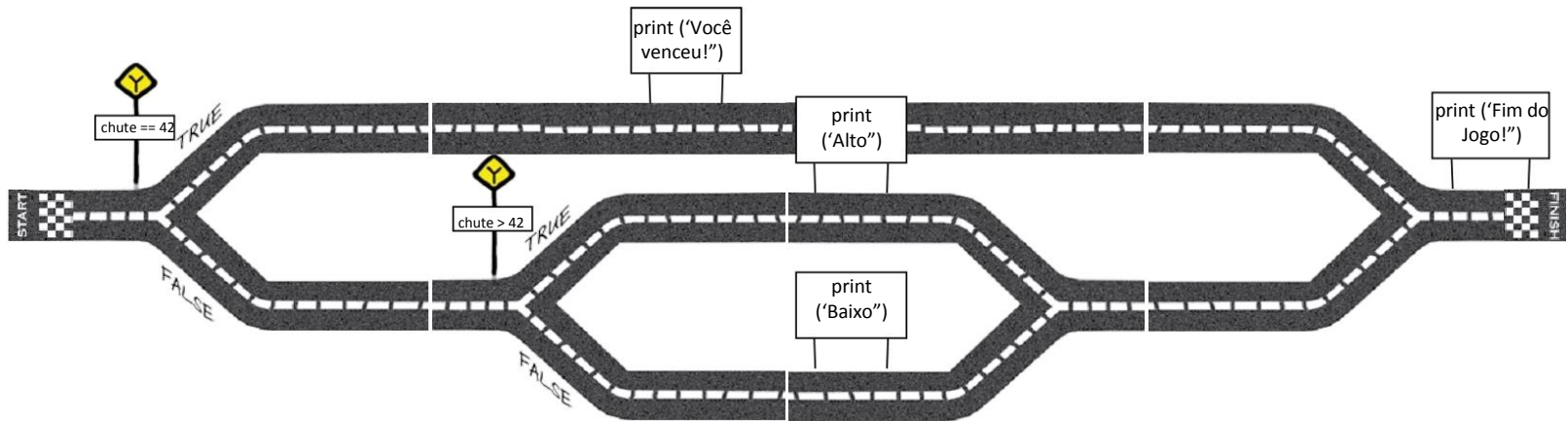




# Dando dicas

- O programa somente diz se acertou ou não
- Para ajudar você dirá “Alto” ou “Baixo” caso a pessoa erre
- Como ficaria a estrada?

# Dando dicas



# Dando dicas

```
print ('Bem vindo!')
g = input ('Chute um número: ')
chute = int(g)
if chute == 42:
    print ('Você venceu!')
else:
    if chute > 42:
        print ('Alto')
    else:
        print ('Baixo')
print ('Fim do jogo!')
```

Utilize BACKSPACE  
e TAB para ir e  
voltar

# Os usuários ainda não gostam



# Repetições

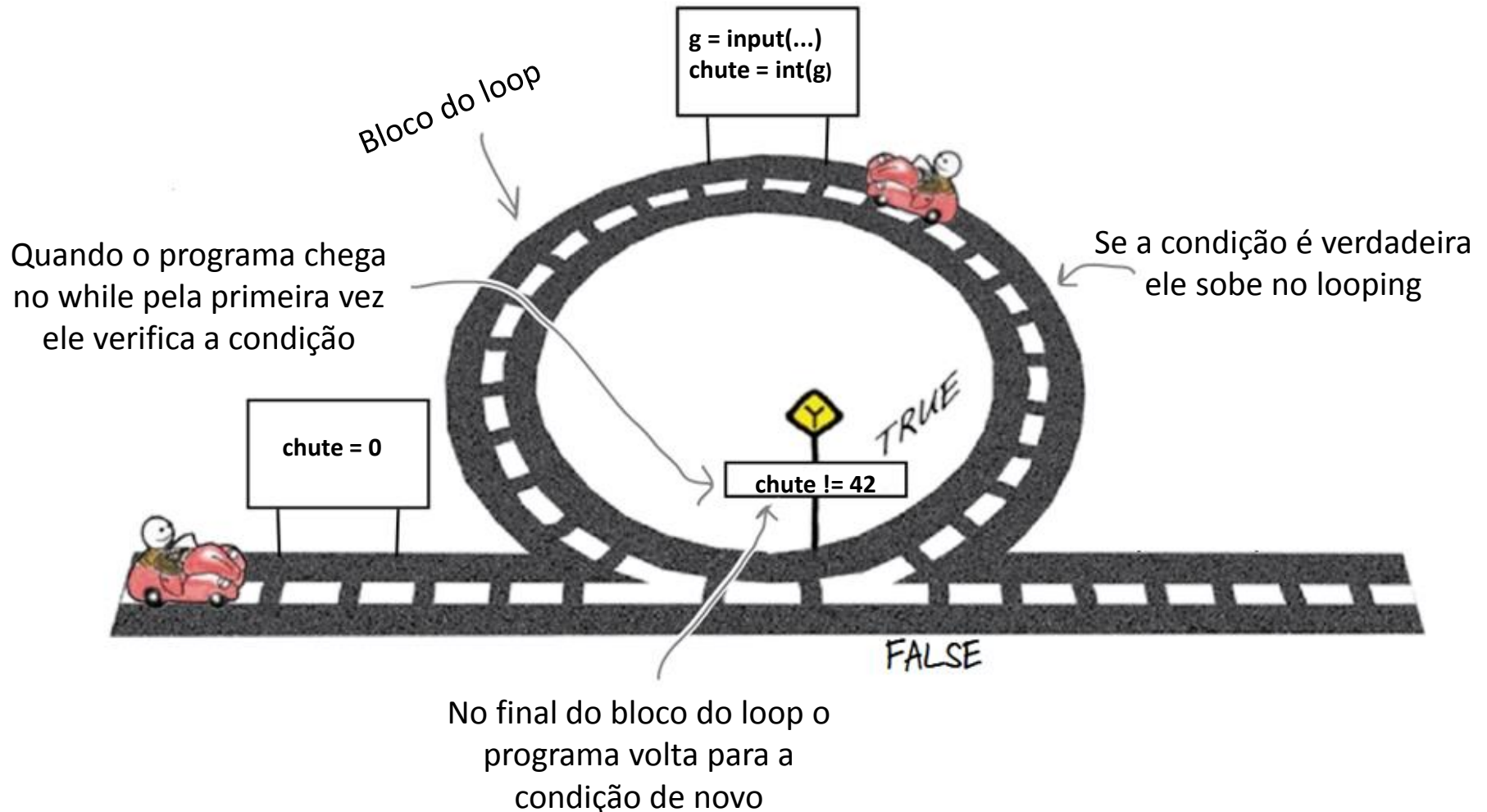
Seria tão legal repetir várias vezes as mesmas linhas de código, mas isso é um sonho...



# Repetições

```
print ('Bem vindo!')  
→ chute = 0  
→ while chute != 42:  
    g = input ('Chute um número: ')  
    chute = int(g)  
    if chute == 42:  
        print ('Você venceu!')  
    else:  
        if chute > 42:  
            print ('Alto')  
        else:  
            print ('Baixo')  
print ('Fim do jogo!')
```

# Repetições



# Os usuários ainda não gostam





# Sortear o número a ser adivinhado



```
from random import randint
print ('Bem vindo!')
sorteado = randint (1, 100)
chute = 0
while chute != sorteado:
    g = input ('Chute um número: ')
    chute = int(g)
    if chute == sorteado:
        print ('Você venceu!')
    else:
        if chute > sorteado:
            print ('Alto')
        else:
            print ('Baixo')
print ('Fim do jogo!')
```

# Agora sim!

Show de bola! Não importa o quanto eu jogue, sempre terei um número novo!



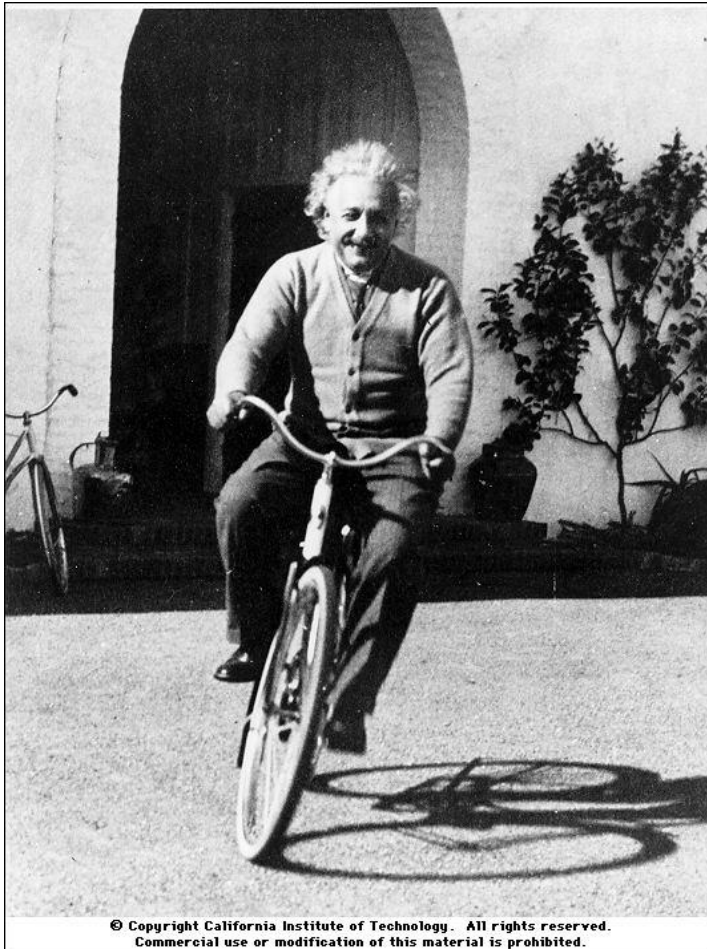
# Resumo

- Você criou um game!
- Comandos fazem coisas
- Desvios decidem coisas
- Laços repetem coisas
- As condições ajudam você a decidir se algo é True ou False
- A atribuição define um nome para um dado

# Ferramentas do Python

- Desvios `if/else`
- Laços `while`
- Operador de atribuição `=`
- Operador de igualdade `==`
- Operador diferente `!=`
- Exibir uma mensagem `print`
- Ler a entrada `input`
- Converter para inteiro `int`
- Sortear um número inteiro `randint`

# Lista VI com testes!



*“A vida é como  
andar de bicicleta.  
Para manter o  
equilíbrio, é preciso  
se manter em  
movimento”.*  
*Einstein.*