

# *Introdução ao Processamento de Dados*

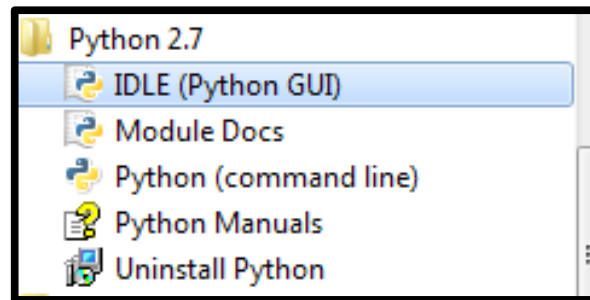
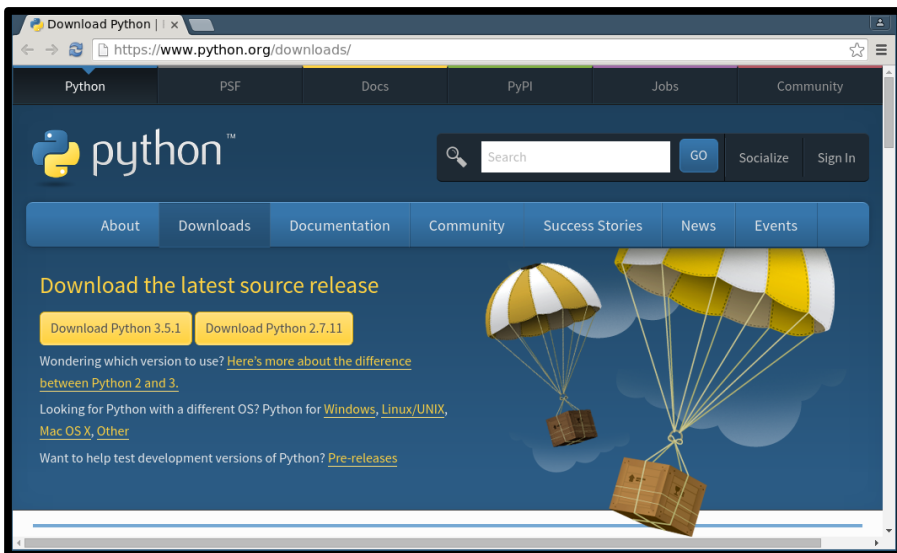
**Francisco Sant'Anna**

`francisco@ime.uerj.br`

`http://github.com/fsantanna/IPD`

# Python

- Download 2.7.\*
  - [www.python.org/downloads](http://www.python.org/downloads)
- Instalar o arquivo baixado
  - <https://www.python.org/ftp/python/2.7.11/python-2.7.11.msi>
- Executar o “IDLE”



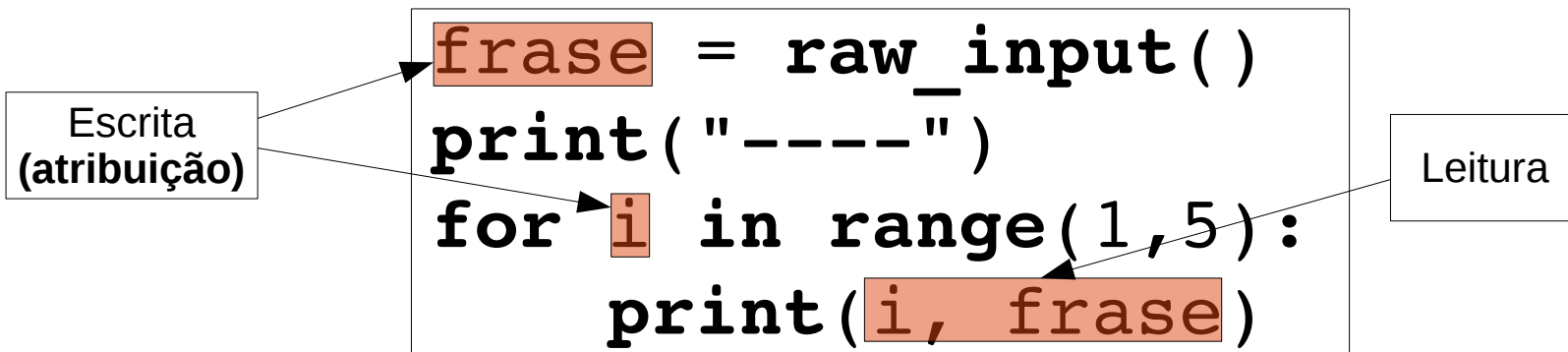
anim.gif

# Conceitos de Programação

- Variável
- Atribuição
- Controle de Fluxo de Execução
  - Sequência
  - Condicional
  - Repetição
- Constante (e.g., 1, "Ola Mundo")
- Expressão (e.g., 1+10, resposta=="nao", raw\_input())

# Variáveis

- Uma abstração da memória do computador
- Uma “etiqueta” que representa uma região de memória
- O programa pode **ler** e **escrever** valores na memória através dessas etiquetas.



# Um jogo de “Adivinha”

Escrita vs Leitura

```
print "Bem-vindo ao jogo..."
guess = raw_input("Adivinhe o numero: ")
if guess == "5":
    print "Voce ganhou!"
else:
    print "Voce perdeu..."
print "Fim"
```

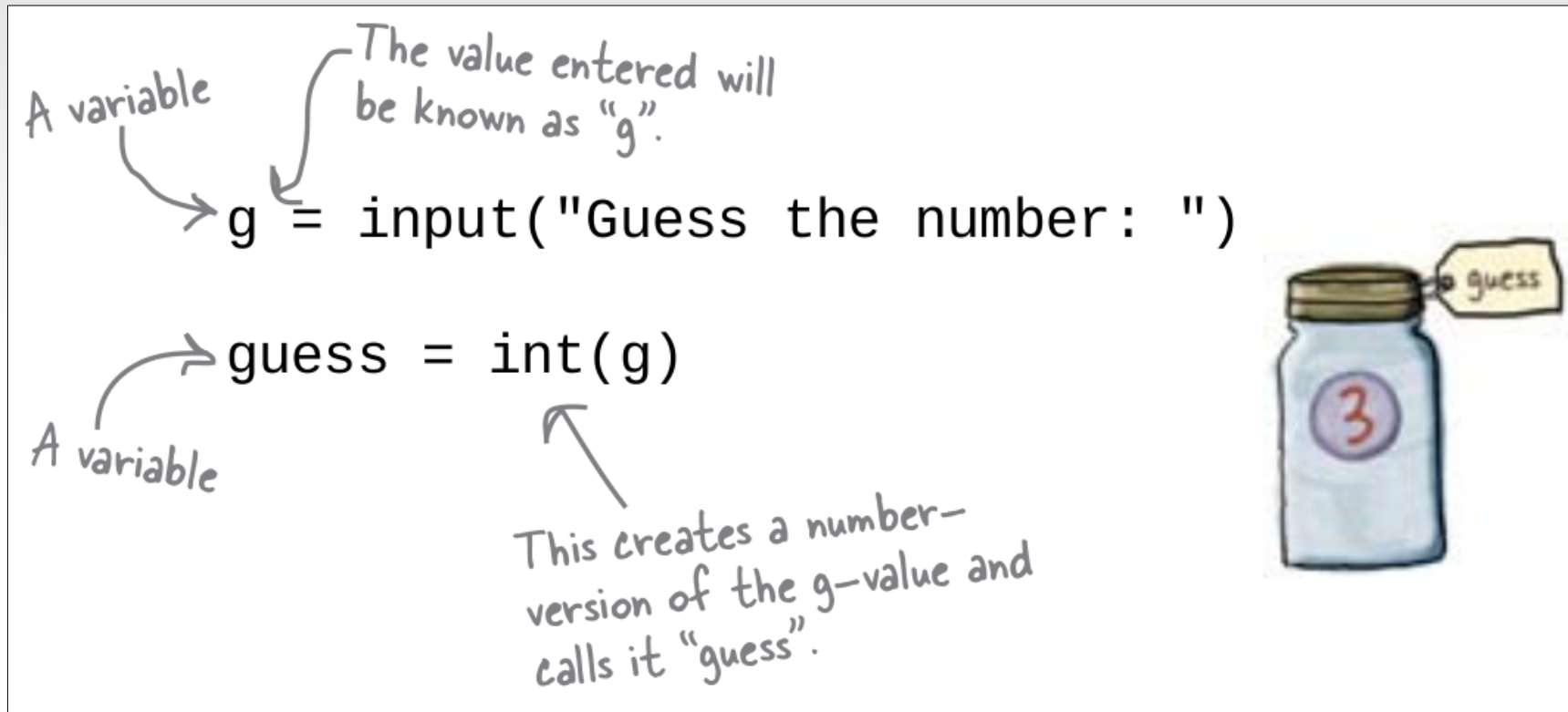
Atribuição vs Comparação

variáveis?

# Um jogo de “Adivinha”

```
print "Bem-vindo ao jogo..."
g = raw_input("Adivinhe o numero: ")
guess = int(g)
if guess == 5:
    print "Voce ganhou!"
else:
    print "Voce perdeu..."
print "Fim"
```

# Variáveis (memória)



Créditos: "Head First Programming"  
(Python 3)

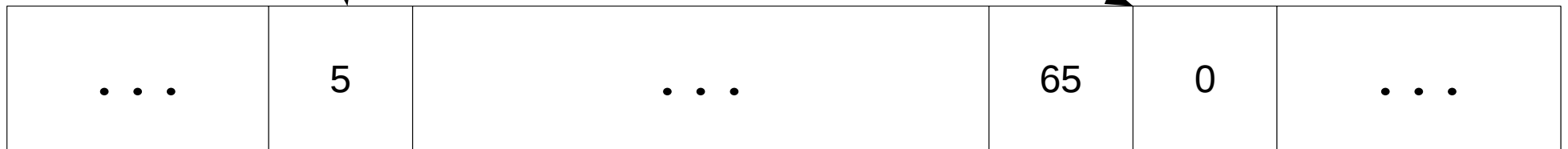
# Variáveis (memória)

```
...  
g = raw_input("Adivinhe o numero: ")  
guess = int(g)  
...
```

escrita

escrita

leitura



guess

g

... memória ...



# Controle de Fluxo

- Sequência
- Condicional
- Repetição

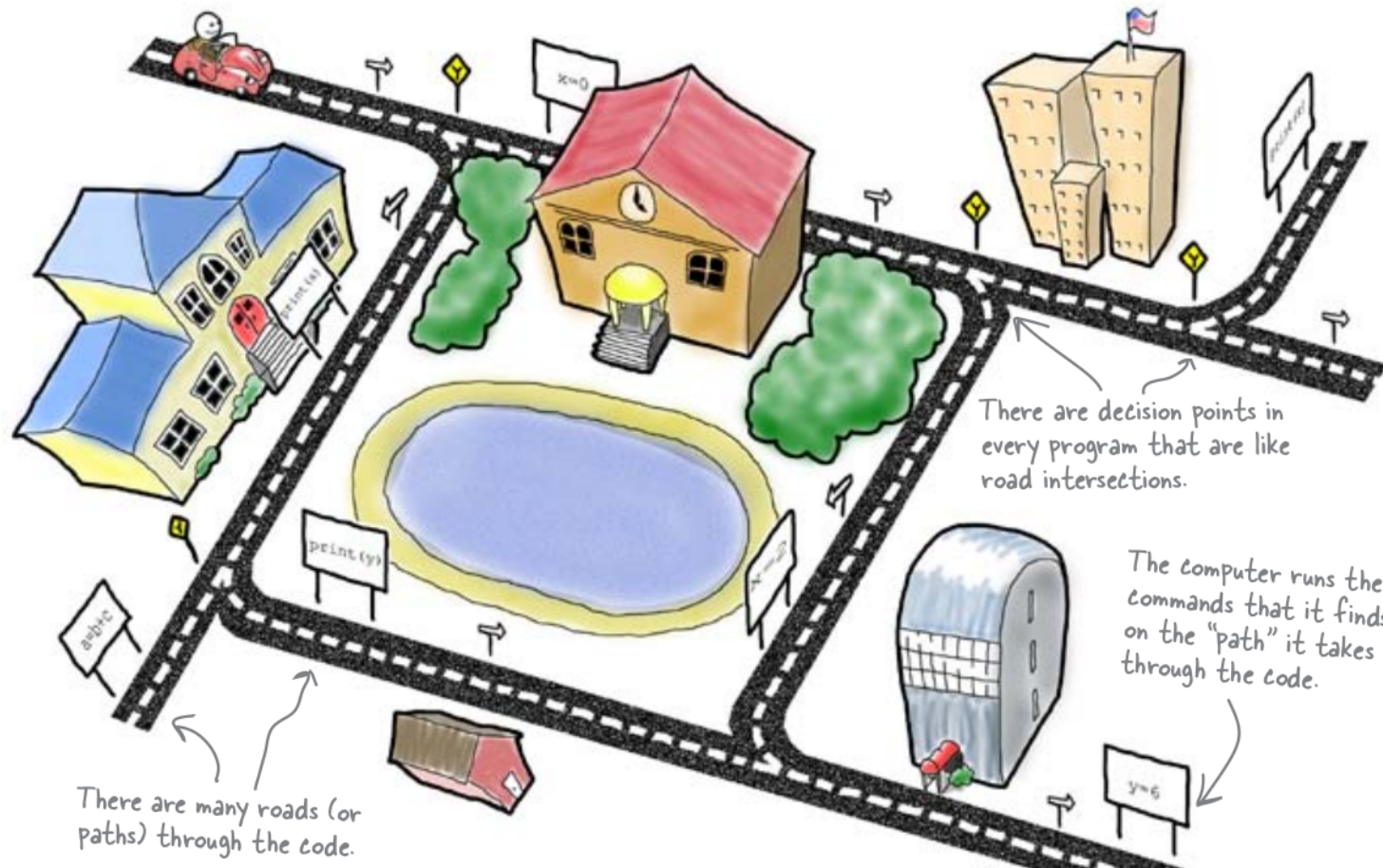
```
frase = raw_input()  
print("----")  
for i in range(1,5):  
    print(i, frase)
```

```
print "Bem-vindo ao jogo..."  
g = raw_input("Adivinhe o numero: ")  
guess = int(g)  
if guess == 5:  
    print "Voce ganhou!"  
else:  
    print "Voce perdeu..."  
print "Fim"
```

# Controle de Fluxo

## Codeville: Your program is like a network of roads

Programs need to do different things under different circumstances. In the game, the code displays “You win!” if the user guesses the number correctly, and “You lose!” if not. This means that all programs, even really simple programs, typically have multiple **paths** through them.

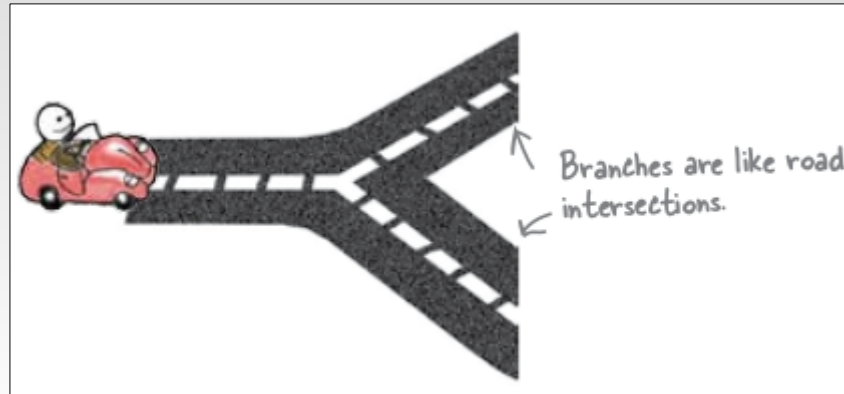


# Sequência (linha, `;`)

```
print "Bem-vindo ao jogo..."
g = raw_input("Adivinhe o numero: ")
guess = int(g)
if guess == 5:
    print "Voce ganhou!"
else:
    print "Voce perdeu..."
print "Fim"
```

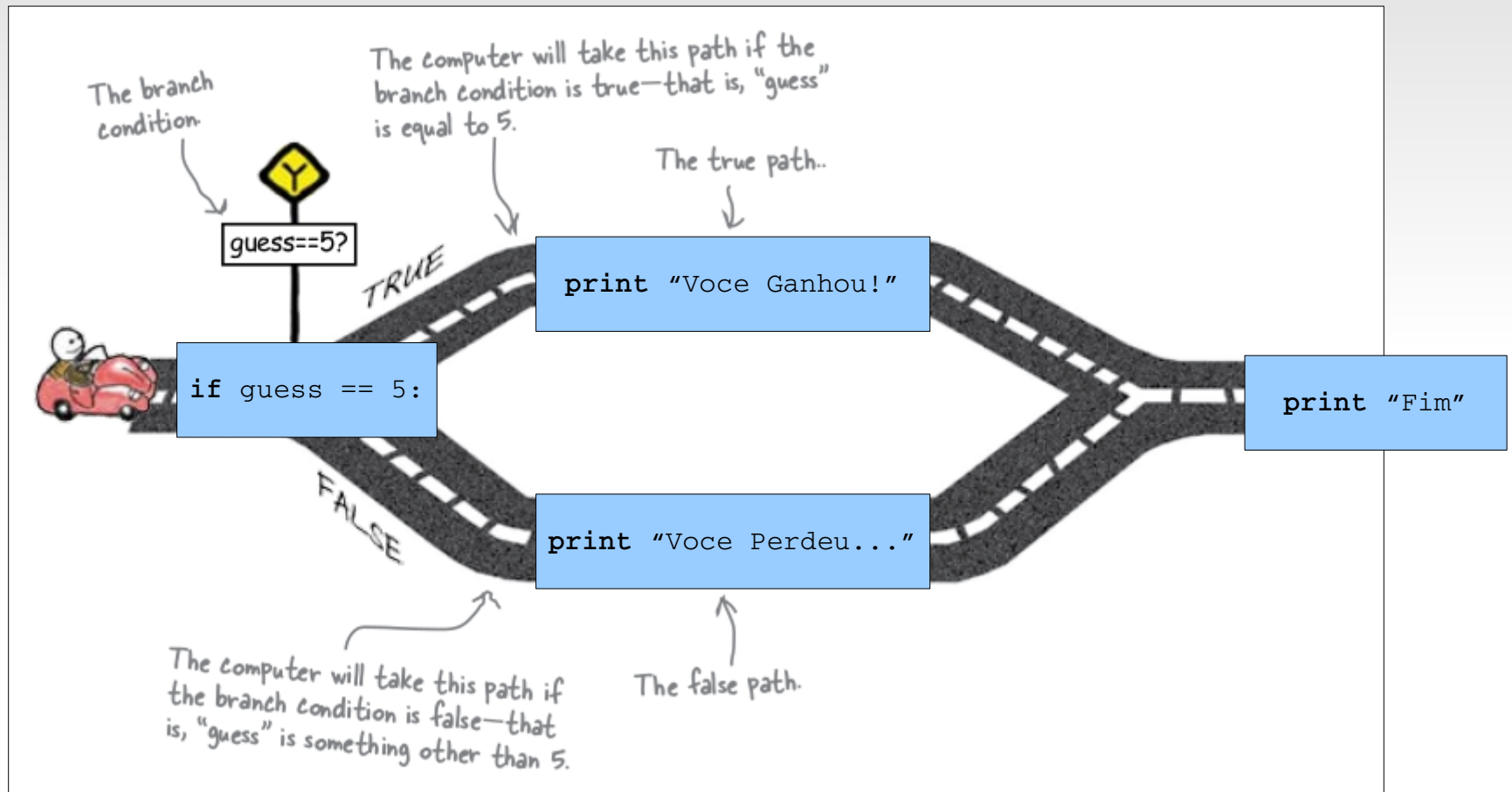
```
print "Bem-vindo ao jogo..."; g = raw_input("Adivinhe o numero: "); guess = int(g)
```

# Condicional (if)



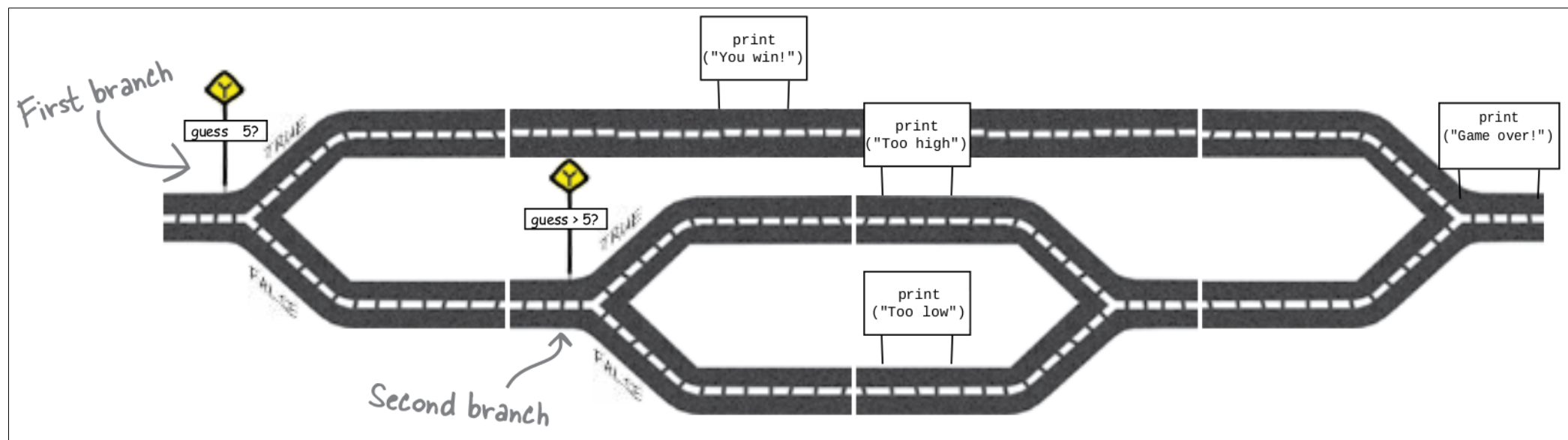
```
print "Bem-vindo ao jogo..."
g = raw_input("Adivinhe o numero: ")
guess = int(g)
if guess == 5:
    print "Voce ganhou!"
else:
    print "Voce perdeu..."
print "Fim"
```

# Condicional (if)



# Exercício: Condicional (if)

- Construir uma “pista” que exiba a mensagem correta
- Não é necessário usar todas as peças



# Repetição (for -> while)

```
frase = raw_input()  
print("----")  
for i in range(1,5):  
    print(i, frase)  
print("----")
```

```
frase = raw_input()  
print("----")  
i = 1  
while i < 5:  
    print(i, frase)  
    i = i + 1  
print("----")
```

```
resposta = "nao"
while resposta == "nao":
    print "Estamos chegando?"
    resposta = raw_input()
print "Chegamos!"
```

