

LÓGICA DE PROGRAMAÇÃO

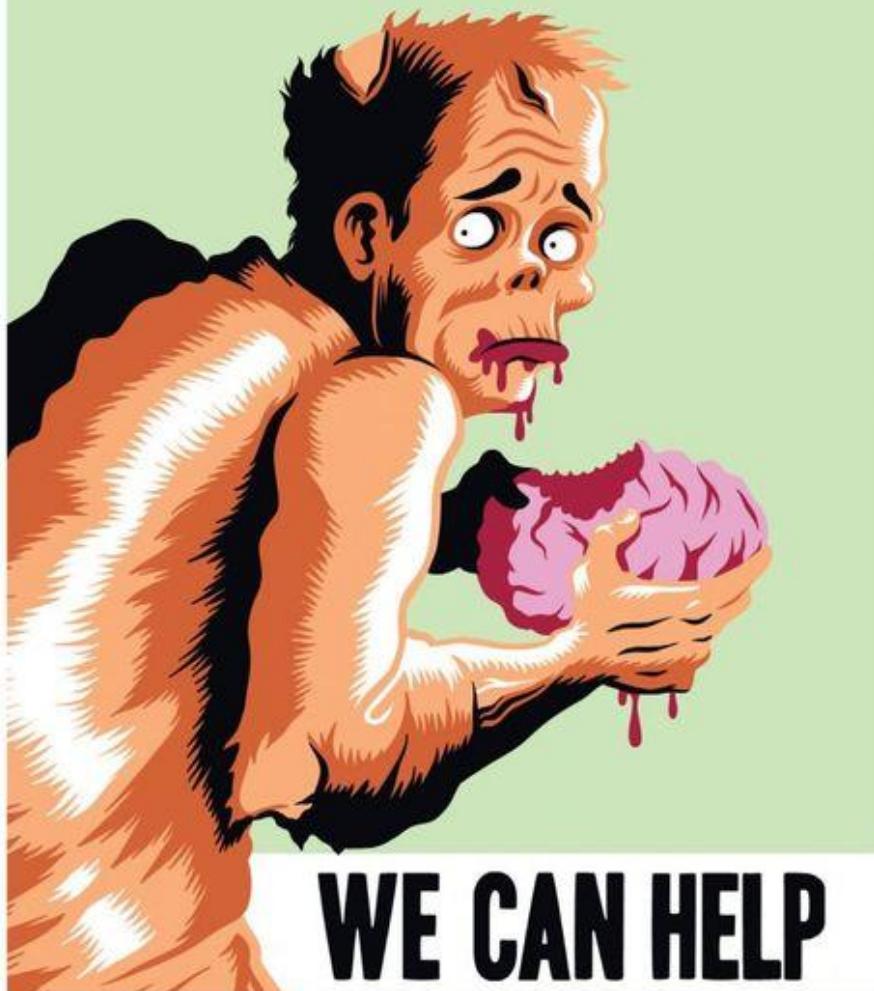
Professor Jefferson Chaves
jefferson.chaves@ifc-araquari.edu.br



Técnico Integrado em
INFORMÁTICA

- Variáveis e entrada de dados;
- Condições;
- Repetições;
- Listas;
- Funções;

STILL EATING BRAINS?

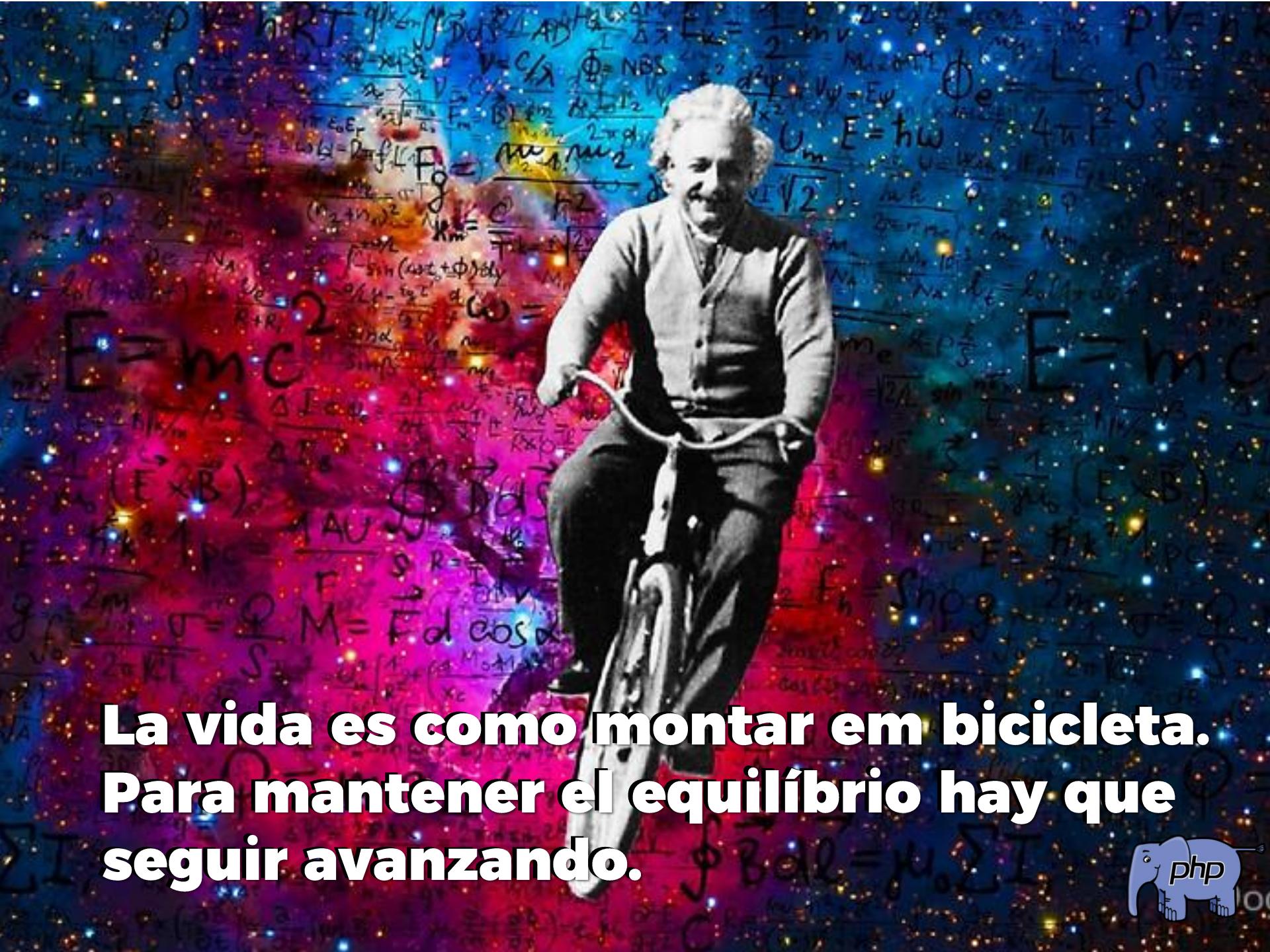


WE CAN HELP

KICK YOUR CEREBRAL CONSUMPTION ADDICTION!
THE DEPARTMENT OF INTEGRATION IS HERE FOR YOU



haves



**La vida es como montar em bicicleta.
Para mantener el equilibrio hay que seguir avanzando.**



- Escreva seus próprios programas!
- Usar o software dos outros, sempre limitará àquilo que outras pessoas acham que você deseja fazer
- Você deseja ser programado ou ser o programador?
- Você pode assumir o controle!



```
echo "digite um número";
$chute = (int) fgets(STDIN);

if($chute == 42){
    echo "você venceu";
} else{
    echo "você perdeu";
}
```



TIPOS DE
DADOS

```
echo "digite um número";
```

```
$chute = (int) fgets(STDIN);
```

```
if($chute == 42){
```

```
    echo "você venceu";
```

```
}else{
```

```
    echo "você perdeu";
```

```
}
```



VARIÁVEIS

```
echo "digite um número";
$chute = (int) fgets(STDIN);

if($chute == 42){
    echo "você venceu";
} else{
    echo "você perdeu";
}
```



CONTROLES

```
echo "digite um número";
$chute = (int) fgets(STDIN);

if($chute == 42){
    echo "você venceu";
} else{
    echo "você perdeu";
}
```



INDENTAÇÃO

LÓGICA DE PROGRAMAÇÃO

OPERADORES LÓGICOS

```
echo "digite um número";
$chute = (int) fgets(STDIN);

if($chute == 42){
    echo "você venceu";
} else{
    echo "você perdeu";
}
```



ATRIBUIÇÃO

LÓGICA DE PROGRAMAÇÃO

OPERADORES LÓGICOS

```
echo "digite um número";
```

```
$chute = (int) fgets(STDIN);
```

```
if($chute == 42){
```

```
    echo "você venceu";
```

```
}else{
```

```
    echo "você perdeu";
```

```
}
```



COMPARAÇÃO

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

- A identação separa blocos de comandos
 - “Cada um no seu quadrado”
- Um igual (=) significa atribuição
 - Ex.: \$chute = (int) fgets(STDIN) (chute recebe o valor inteiro digitado) .
- Dois iguais (==) significa comparação:
 - Ex.: chute == 42 (chute é igual a 42?)

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

- Como tratar os erros?
 - Sintáticos: muita atenção e prática
 - Semânticos: teste de mesa ou simulação

Um programa é mais que uma lista de comandos

```
print("bem vindo ao meu programa");  
print("Volte sempre");
```



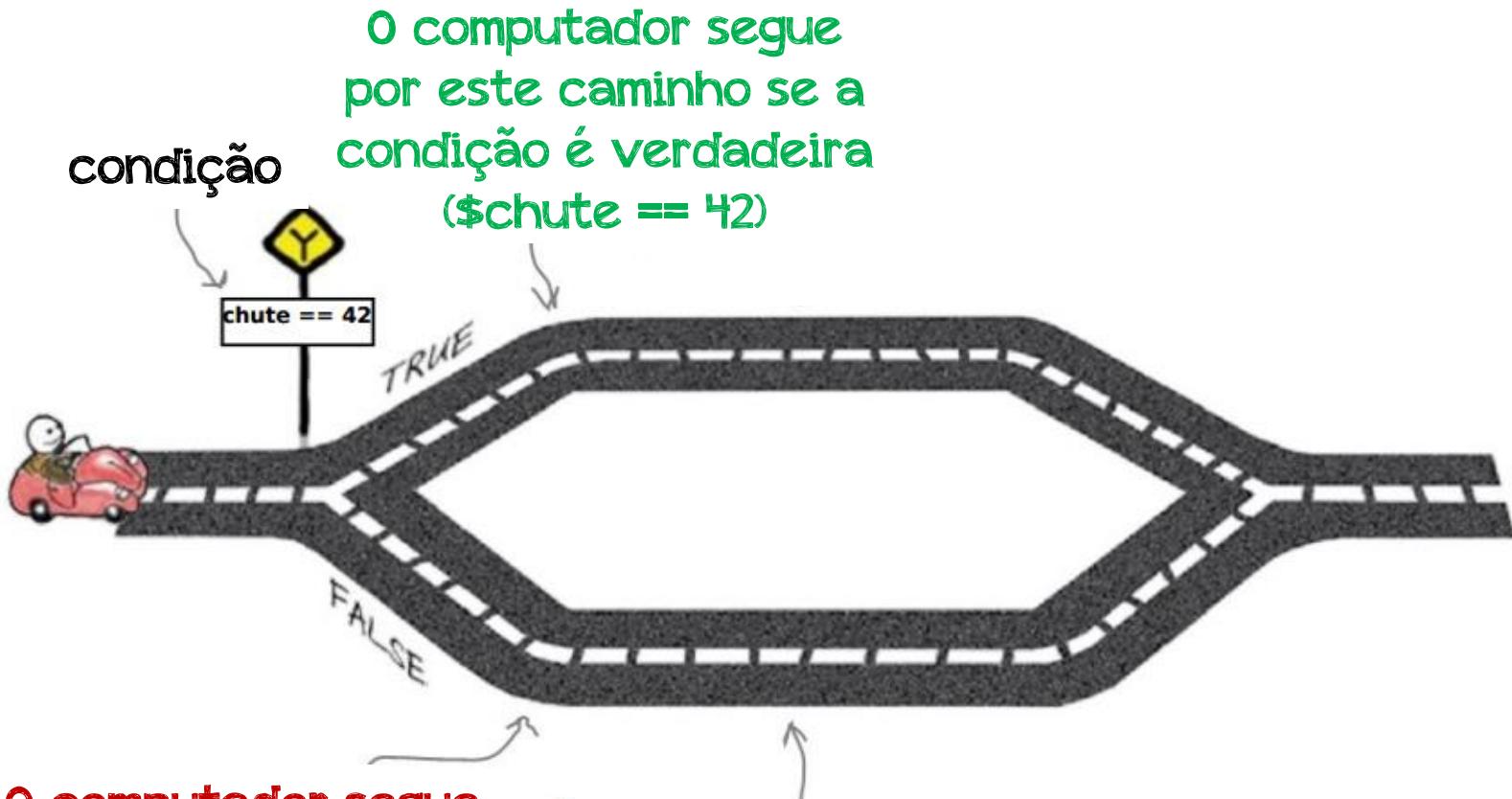
PROGRAMAÇÃO

REVISÃO



PROGRAMAÇÃO

REVISÃO

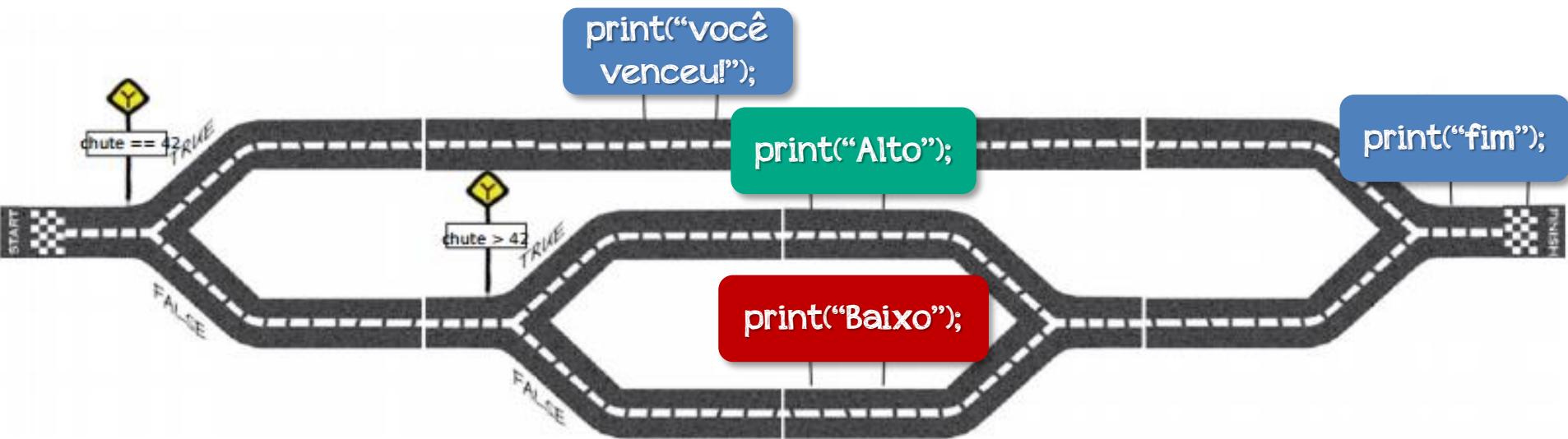


O computador segue por este caminho se a condição é false
(\$chute != 42)

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

PROGRAMAÇÃO

REVISÃO



```
echo "digite um número";
$chute = (int) fgets(STDIN);

if($chute == 42){
    echo "você venceu";
} else{
    if($chute > 42){
        echo "alto";
    } else{
        echo "baixo";
    }
}
```

PROGRAMAÇÃO

REVISÃO



Jefferson de Oliveira Chaves

PROGRAMAÇÃO

REVISÃO

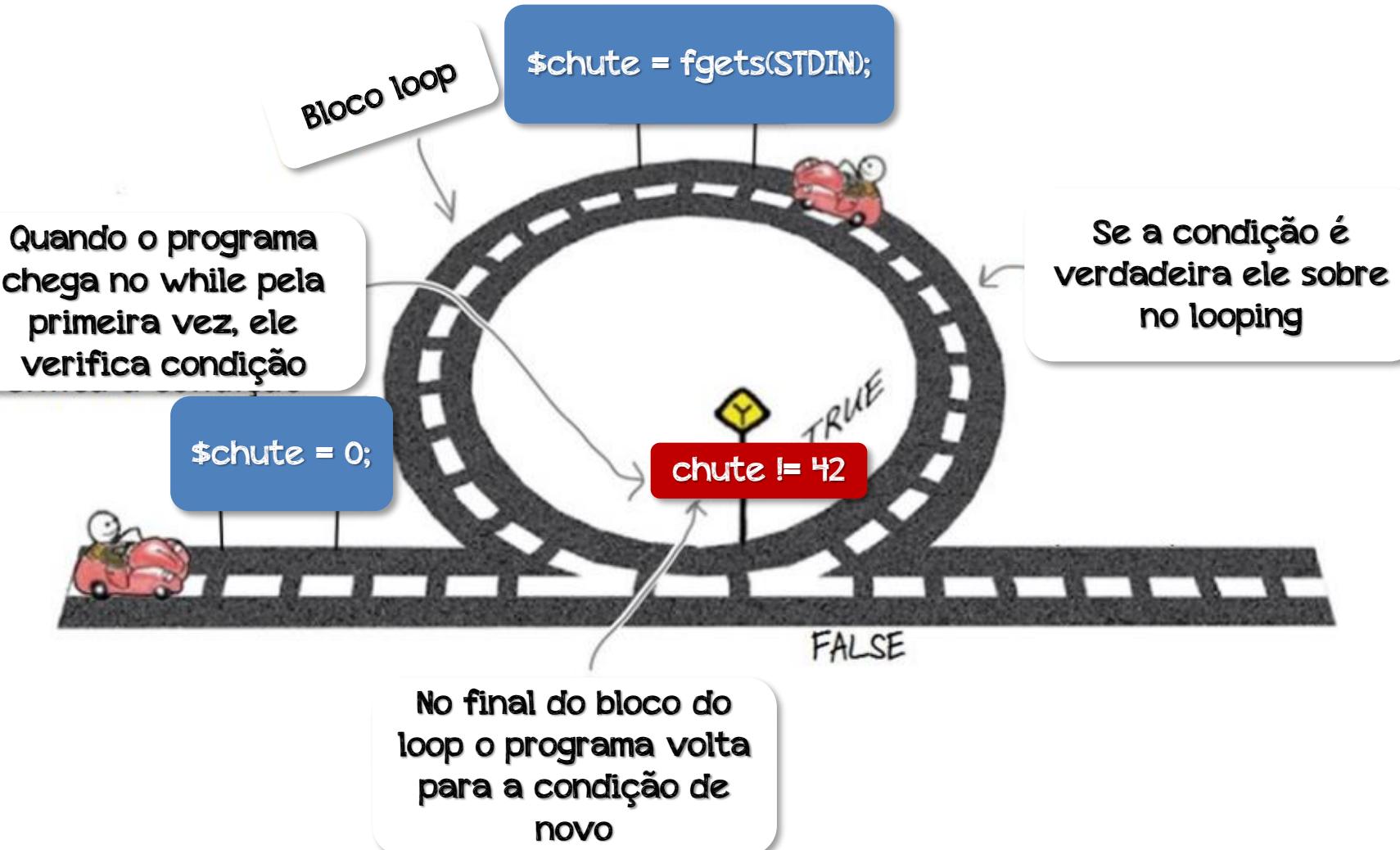


Jefferson de Oliveira Chaves

```
$chute = 0;  
while($chute != 42){  
    echo "digite um número";  
    $chute = (int) fgets(STDIN);  
  
    if($chute == 42){  
        echo "você venceu";  
    }else{  
        if($chute > 42){  
            echo "alto";  
        }else{  
            echo "baixo";  
        }  
    }  
}
```

PROGRAMAÇÃO

REPETIÇÕES



PROGRAMAÇÃO

REVISÃO



Jefferson de Oliveira Chaves

PRGRAMAÇÃO

DOCUMENTAÇÃO

NÚMEROS RANDÔMICOS E A DOCUMENTAÇÃO DO PHP;

PROGRAMAÇÃO

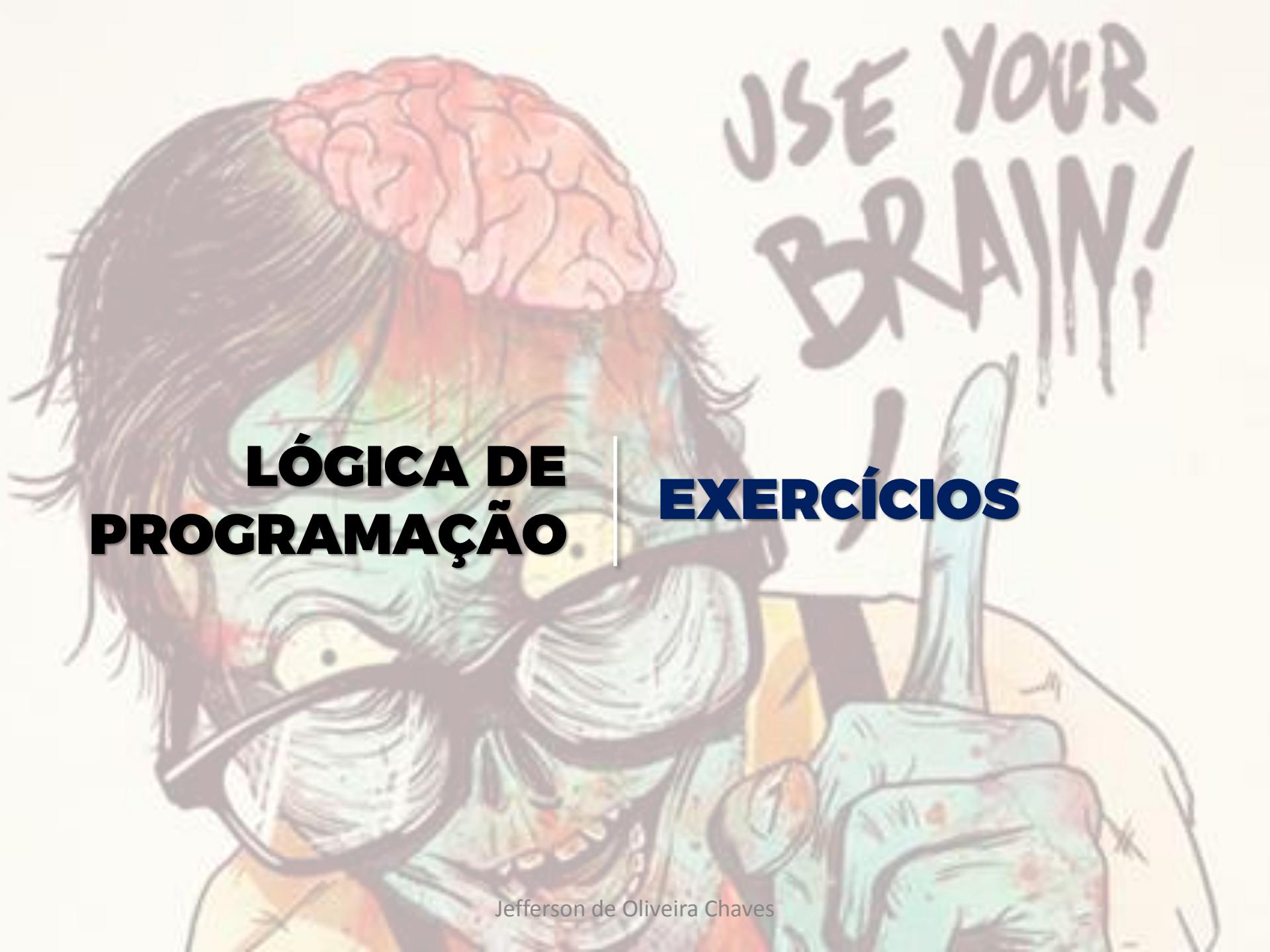
REVISÃO

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



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

- **FUNÇÕES?**
 - Definição;
 - Quais as vantagens?
 - Chamada;
 - Parâmetros;
 - Retorno;
 - Escopo;



LÓGICA DE PROGRAMAÇÃO

EXERCÍCIOS

Jefferson de Oliveira Chaves

1. Palavra palindroma. Construa uma função que receba uma string como parâmetro e retorne true ou false caso ela seja palindroma.

- arara -> arara : retorna true;
- bolacha -> ahcalob : retorna false;

2. **Embaralha palavra.** Construa uma função que receba uma string como parâmetro e devolva outra string com os caracteres embaralhados. Por exemplo: se função receber a palavra *raposa*, pode retornar *opsara*, *prsaoa* ou qualquer outra combinação possível, de forma aleatória. Padronize em sua função que todos os caracteres serão devolvidos em caixa alta ou caixa baixa, independentemente de como foram digitados.

3. Data com mês por extenso. Construa uma função que receba uma data no formato *DD/MM/AAAA* e devolva uma string no formato *D de mesPorExtenso de AAAA*.

4. **Conversão de horas.** Faça um programa que converta da notação de 24 horas para a notação de 12 horas. Por exemplo:
 - converter 14:25 para 2:25 PM

Deve haver pelo menos duas funções: uma para fazer a conversão (`converteHoras`) e uma para a saída (`exibeHoras`).

Inclua um loop que permita que o usuário repita esse cálculo para novos valores de entrada todas as vezes que desejar.

5. Século do ano. Faça uma função que receba um ano como parâmetro e retorne o século deste ano. Exemplos:

- 2000 retorna 20;
- 2001 retorna 21;
- 1740 retorna 18