



Centro Paula Souza



FATEC PRESIDENTE PRUDENTE

DISCIPLINA: ELETIVA I

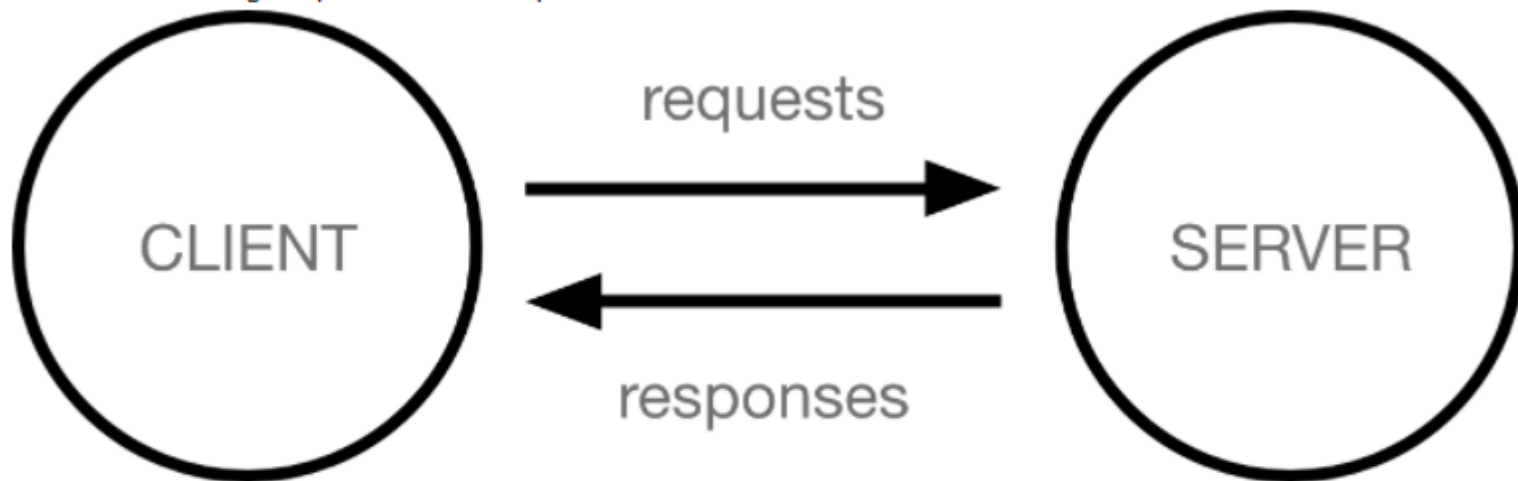
AULA: INTRODUÇÃO A CONCEITOS DE SISTEMAS WEB EM PHP

SISTEMAS LOCAIS X SISTEMAS WEB



COMO FUNCIONA A WEB

Computadores conectados à web são chamados **clientes** e **servidores**. Um diagrama simplificado de como eles interagem pode ter essa aparência:



- Clientes são os típicos dispositivos conectados à internet dos usuários da web (por exemplo, seu computador conectado ao seu Wi-Fi ou seu telefone conectado à sua rede móvel) e programas de acesso à Web disponíveis nesses dispositivos (geralmente um navegador como Firefox ou Chrome).
- Servidores são computadores que armazenam páginas, sites ou aplicativos. Quando o dispositivo de um cliente quer acessar uma página, uma cópia dela é baixada do servidor para a máquina do cliente para ser apresentada no navegador web do usuário.

COMO FUNCIONA A WEB

- TCP/IP: Protocolo de Controle de Transmissão e Protocolo de Internet (Transmission Control Protocol e Internet Protocol) são protocolos de comunicação que definem como os dados trafegam pela web.
- HTTP: Protocolo de Transferência de Hypertexto (Hypertext Transfer Protocol) é um protocolo de aplicação que define uma linguagem para clientes e servidores se comunicarem entre si.
- DNS: Servidor de Nome de Domínio (Domain Name Servers) são como um catálogo de endereços para sites. Quando você digita um endereço web no seu navegador, o navegador procura no servidor de DNS para localizar o endereço real do site, antes que ele possa recuperar o site.

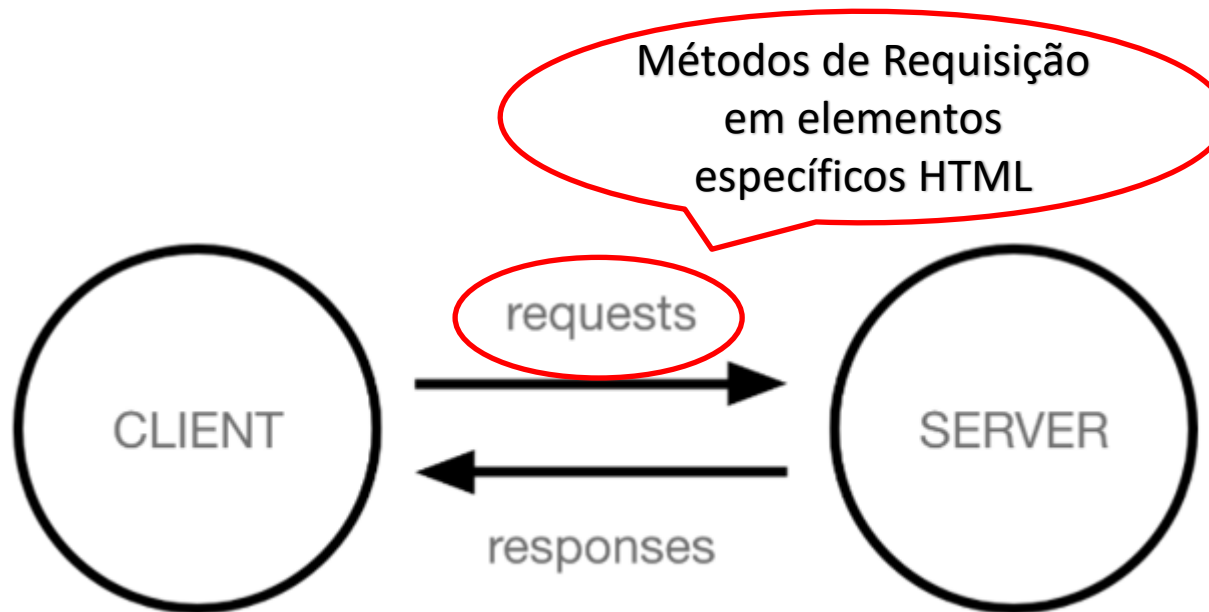
COMO FUNCIONA A WEB

1. O navegador vai para o servidor de DNS e encontra o endereço verdadeiro de onde o site está hospedado
2. O navegador manda uma mensagem de requisição HTTP para o servidor, pedindo que envie uma cópia do site ao cliente
3. O servidor começa a enviar os arquivos do site para o navegador como uma série de pequenos pedaços chamados pacotes de dados
4. O navegador monta os pequenos pedaços em um site completo e o mostra a você

TECNOLOGIAS PARA DESENVOLVIMENTO WEB

- HTML
 - CSS
 - JAVASCRIPT
-
- Material complementar: Revisão HTML, CSS e Bootstrap

COMO FUNCIONA A WEB



LINGUAGEM PHP

- O PHP é uma linguagem que permite criar sites WEB dinâmicos, possibilitando uma interação com o usuário através de formulários e parâmetros da URL.
- O código PHP é executado no servidor, sendo enviado para o cliente apenas HTML puro.
- Desta maneira é possível interagir com bancos de dados e aplicações existentes no servidor, com a vantagem de não expor o código fonte para o cliente.

CARACTERÍSTICAS DA LINGUAGEM

- É uma linguagem de fácil aprendizado;
- Tem suporte a um grande número de bancos de dados como: dBase, Interbase, SQLServer, MySql, Oracle, Sybase, PostgreSQL e vários outros. É multiplataforma, tendo suporte aos sistemas Operacionais mais utilizados no mercado;
- Seu código é livre, não é preciso pagar por sua utilização e pode ser alterado pelo usuário na medida da necessidade de cada usuário
- Não precisa ser compilado.

FUNCIONAMENTO



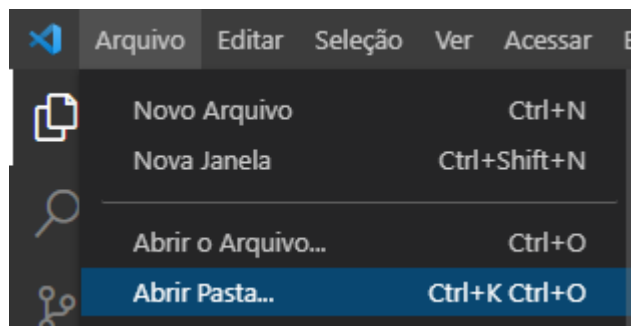
BOAS PRÁTICA

- PHP Framework Interop Group (PHP-FIG) é um grupo formado por desenvolvedores da comunidade PHP que definem recomendações a serem utilizadas em aplicações PHP
- Inicialmente aplicável apenas para Frameworks
- PSR-1 a PSR-4 estão em vigor (PSR-2 substituído por PSR-12)

CRIANDO O PRIMEIRO PROJETO EM PHP

- Configurar repositório no Git e GitHub

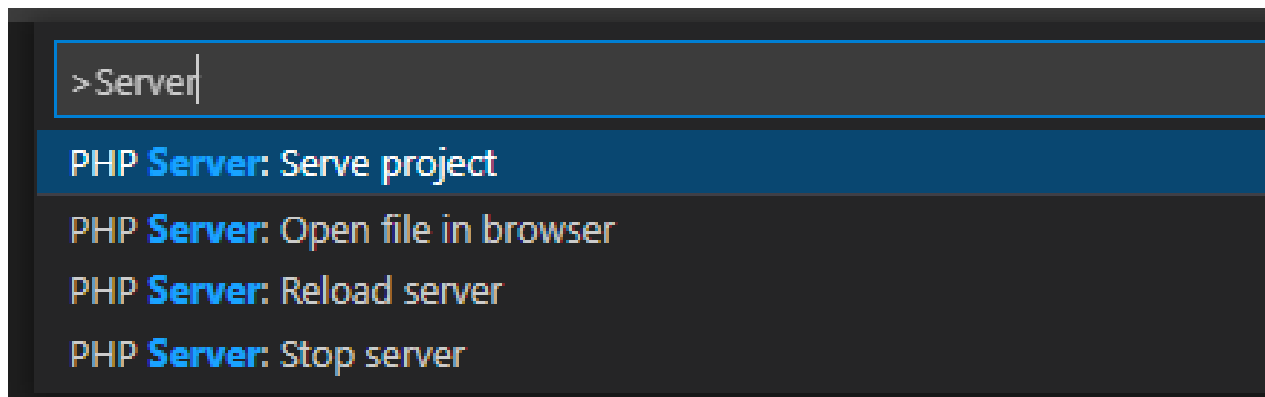
CRIANDO O PRIMEIRO PROJETO EM PHP



```
index.php > html
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <!-- Required meta tags -->
5     <meta charset="pt-br">
6     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7
8     <!-- Bootstrap CSS -->
9     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
10
11     <title>Olá mundo!</title>
12   </head>
13   <body>
14     <h1>Olá! Hoje é <?php echo date("d/m/Y"); ?></h1>
15
```

CRIANDO O PRIMEIRO PROJETO EM PHP

- Abrir a Paleta de Comando (Ctrl+Shift+P)



CRIANDO O PRIMEIRO PROJETO EM PHP

✖ phpcs: Request workspace/configuration failed with message: Unable t...

EXEMPLO

Delimitadores de código PHP

```
<?php ... ?>
```

```
<?= ... ?>
```

Variáveis são precedidas do caractere \$

- camelCase
- não iniciar variáveis com números ou caracteres especiais

Comentários

```
//
```

```
#
```

```
/* ... */
```

Comandos de saída

```
echo
```

```
print()
```

EXEMPLO

Diferença do uso de apóstrofo e
aspas duplas em funções de saída

Operador `.` para concatenação

Constantes
`define("CONSTANTE", 100);`

URL

- O alvo de uma requisição HTTP é chamada de "resource", ou recurso em português, com a natureza ainda não definida; Isso pode ser um documento, uma foto ou qualquer outra coisa. Cada recurso é identificado por uma Identificação de recursos uniforme, do inglês Uniform Resource Identifier (URI) usada pelo HTTP para identificar recursos.
- A identidade e a localização de recursos na Web são fornecidas, principalmente por um único URL (Uniform Resource Locator, um tipo de URI).

URL

Esquema ou protocolo

http://www.example.com:80/path/

Protocol

Autoridade

http://www.example.com:80/path/to/my

Domain Name

Porta

com:80/path/to/myfile.html?key1=value1

Port

Caminhos

n:80/path/to/myfile.html?key1=value1/

Path to the file

Query / Parâmetros

html?key1=value1&key2=value2#

Parameters

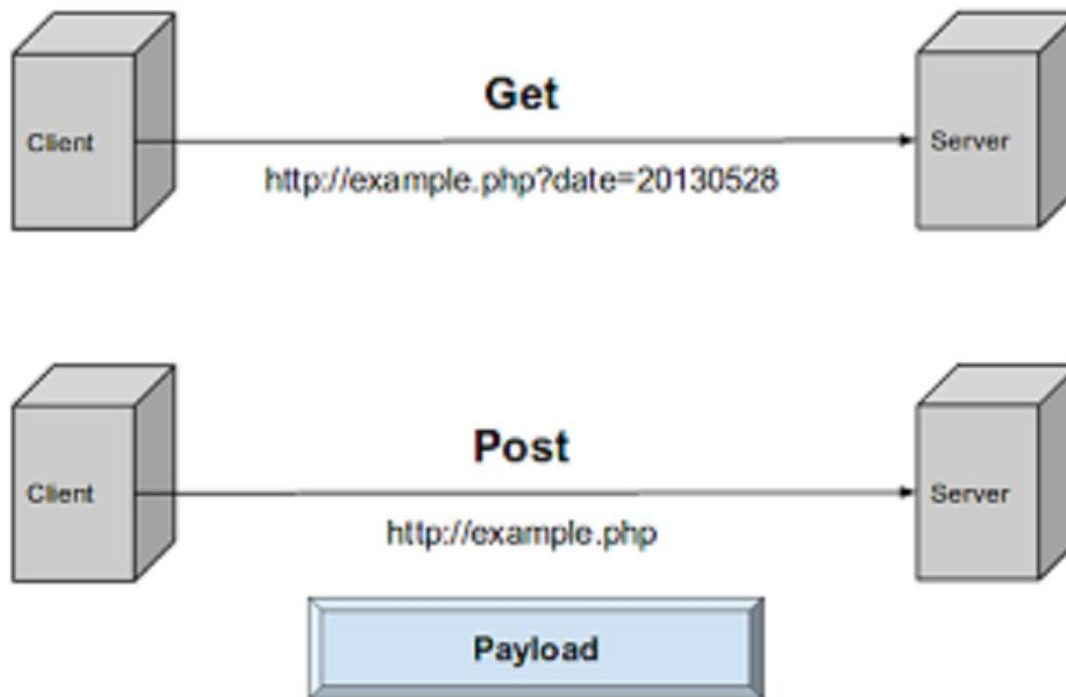
VARIÁVEIS PRÉ-DEFINIDAS

- Variáveis nativas que estão sempre disponíveis em todos escopos da sua aplicação
 - \$_SERVER — Informação do servidor e ambiente de execução
 - \$_GET — variáveis HTTP GET
 - \$_POST — variáveis HTTP POST
 - \$_FILES — utilizada para recuperar dados de arquivos enviados ao servidor
 - \$_SESSION — Variáveis de sessão
 - \$_COOKIE — Variáveis de cookies

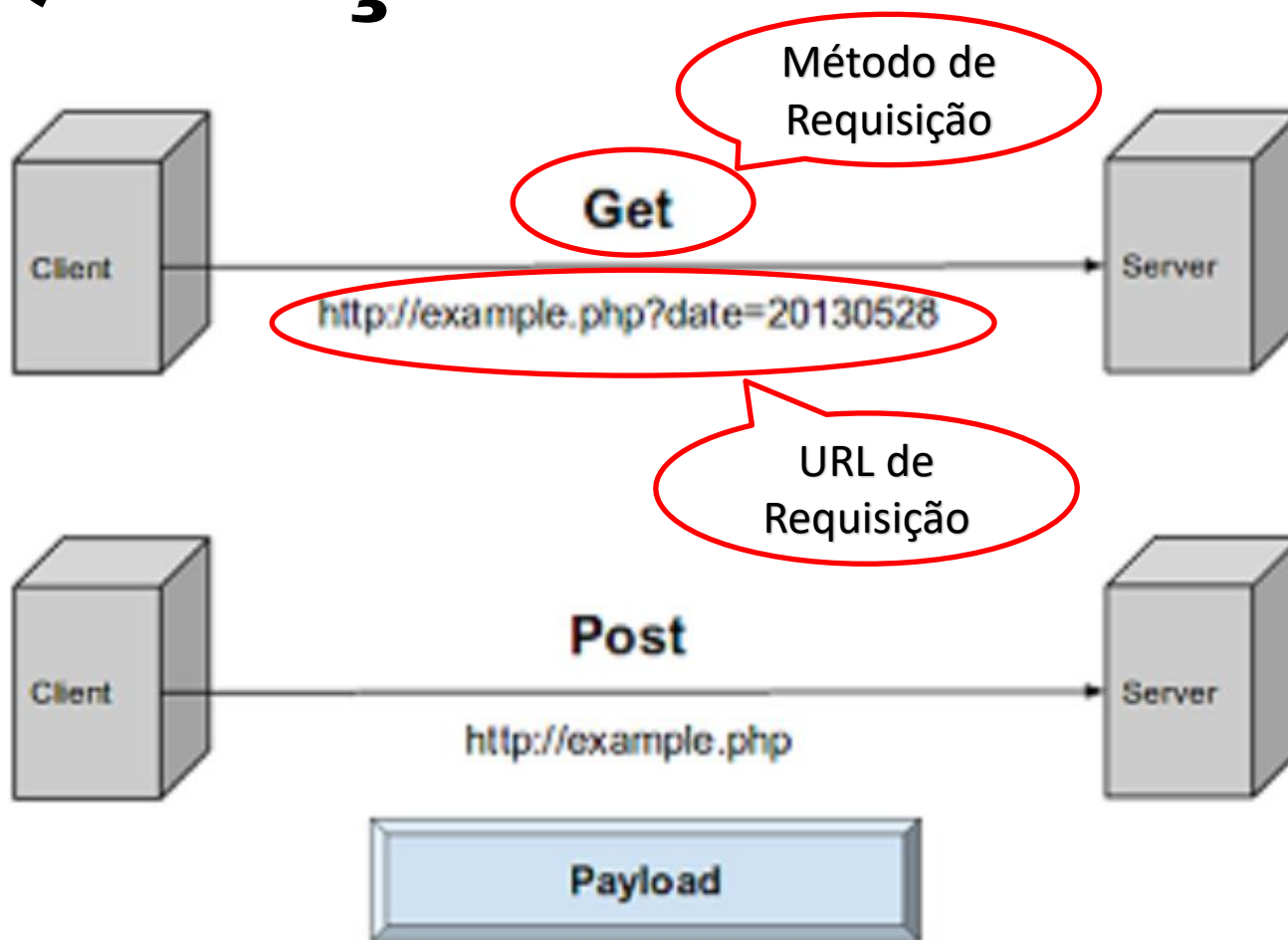
MÉTODOS DE REQUISIÇÃO HTTP

- O protocolo HTTP define um conjunto de métodos de requisição responsáveis por indicar a ação a ser executada para um dado recurso.
- Cada um deles implementa uma semântica diferente
 - O método GET solicita a representação de um recurso específico. Requisições utilizando o método GET devem retornar apenas dados.
 - O método POST é utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor.

MÉTODOS DE REQUISIÇÃO HTTP



REQUISIÇÃO HTTP



VARIÁVEIS PRÉ-DEFINIDAS

- `$_SERVER`
 - `$_SERVER['REQUEST_METHOD']` — Contém o método de request utilizando para acessar a página.
 - `$_SERVER['PATH_INFO']` — caminho real do script relativo ao sistema de arquivos

EXEMPLO

Formulário: envio de requisições com métodos GET e POST

Uso de âncora para requisições com método GET

Uso das variáveis pré-definidas

TIPAGEM

- Faça uma pesquisa: Quais as diferenças entre as tipagens estática, dinâmica, forte e fraca?

TIPAGEM

- Tipos Primitivos

```
string $nome;  
int $quantidade;  
float $valor;  
bool $entregar;
```

As palavras-chave
(e constantes)
“true”, “false” e
“null” DEVEM ser
escritas com letras
minúsculas

TIPAGEM

- Anuláveis

```
?string $nome;  
?int $quantidade;  
?float $valor;  
?bool $entregar;
```

TIPAGEM

- Callables (executáveis ou funções?)

```
$double = function($a) {  
    return $a * 2;  
};
```

Esta é uma função anônima (closure), ou seja, é uma função sem nome especificado

OPERADORES DE ATRIBUIÇÃO

```
<?php
$var += 5;    // Soma 5 em $var;
$var -= 5;    // Subtrai 5 em $var;
$var *= 5;    // Multiplica $var por 5;
$var /= 5;    // Divide $var por 5;
```

Operadores	Descrição
++\$a	Pré-incremento. Incrementa \$a em um e, então, retorna \$a.
\$a++	Pós-incremento. Retorna \$a e, então, incrementa \$a em um.
--\$a	Pré-decremento. Decrementa \$a em um e, então, retorna \$a.
\$a--	Pós-decremento. Retorna \$a e, então, decrementa \$a em um.

OPERADORES ARITMÉTICOS

Operadores	Descrição
+	Adição.
-	Subtração.
*	Multiplicação.
/	Divisão.
%	Módulo (resto da divisão).

OPERADORES RELACIONAIS

Comparadores	Descrição
==	Igual. Resulta verdadeiro (TRUE) se expressões forem iguais.
===	Idêntico. Resulta verdadeiro (TRUE) se as expressões forem iguais e do mesmo tipo de dados.
!= ou <>	Diferente. Resulta verdadeiro se as variáveis forem diferentes.
<	Menor.
>	Maior que.
<=	Menor ou igual.
>=	Maior ou igual.

OPERADORES LÓGICOS

Operador	Descrição
(\$a and \$b)	E: Verdadeiro (TRUE) se tanto \$a quanto \$b forem verdadeiros.
(\$a or \$b)	OU: Verdadeiro (TRUE) se \$a ou \$b forem verdadeiros.
(\$a xor \$b)	XOR: Verdadeiro (TRUE) se \$a ou \$b forem verdadeiros, de forma exclusiva.
(! \$a)	NOT: Verdadeiro (TRUE) se \$a for FALSE.
(\$a && \$b)	E: Verdadeiro (TRUE) se tanto \$a quanto \$b forem verdadeiros.
(\$a \$b)	OU: Verdadeiro (TRUE) se \$a ou \$b forem verdadeiros.

Observação: or e and têm precedência maior que && ou ||.

ESTRUTURAS DE CONTROLE

```
<?php
```

```
if ($expr1) {  
    // if body  
} elseif ($expr2) {  
    // elseif body  
} else {  
    // else body;  
}
```

ESTRUTURAS DE CONTROLE

```
<?php
```

```
switch ($expr) {  
    case 0:  
        echo 'First case, with a break';  
        break;  
    case 1:  
        echo 'Second case, which falls through';  
        // no break  
    case 2:  
    case 3:  
    case 4:  
        echo 'Third case, return instead of break';  
        return;  
    default:  
        echo 'Default case';  
        break;  
}
```

OPERADOR TERNÁRIO

```
$variable = $foo ? 'foo' : 'bar';
```

```
$variable = $foo ?: 'bar';
```

ESTRUTURAS DE REPETIÇÃO

```
<?php
```

```
while ($expr) {  
    // structure body  
}
```

```
<?php
```

```
while (  
    $expr1  
    && $expr2  
) {  
    // structure body  
}
```

ESTRUTURAS DE REPETIÇÃO

```
<?php
```

```
do {  
    // structure body;  
} while ($expr);
```

```
<?php
```

```
do {  
    // structure body;  
} while (  
    $expr1  
    && $expr2  
);
```

ESTRUTURAS DE REPETIÇÃO

```
<?php
```

```
for ($i = 0; $i < 10; $i++) {  
    // for body  
}
```


LISTA DE EXERCÍCIOS 1

1. Faça um programa que receba um valor pago, um segundo valor que é o preço do produto e retorne o troco a ser dado.
2. Faça um programa que receba o valor do quilo de um produto e a quantidade de quilos do produto consumida, calculando o valor final a ser pago.
3. Desenvolva um programa que receba um valor digitado pelo usuário e imprima o texto "o valor é maior que 10" caso isso seja verdade, senão imprima "o valor é menor que 10".
4. Desenvolva um programa que receba um valor digitado pelo usuário e verifique se esse valor é positivo, negativo ou igual a zero. Imprima na tela: "Valor Positivo", "Valor Negativo", "Igual a Zero".
5. Desenvolva um programa que receba quatro notas de um aluno, calcule e imprima a média aritmética das notas e a mensagem de aprovado para média superior ou igual a 7.0 ou a mensagem de reprovado para média inferior a 7.0



Obrigada