



INSTITUTO FEDERAL
PARANÁ

WEB II

Professor: Rodrigo Viecheneski
Email: rodrigo.ifpr.irati@gmail.com



INSTITUTO FEDERAL
PARANÁ

Apresentações:

- Nome:
- Cidade:
- Idade:
- Expectativa com a disciplina:
- Trabalha? Onde? O que faz?



INSTITUTO FEDERAL
PARANÁ

Ementa:

- Desenvolvimento de aplicações web:
 - interface desenvolvida com tecnologias client-side;
 - lógica de negócio e acesso a banco de dados desenvolvidos com tecnologias server-side.
- Desenvolvimento de aplicações em camadas (design pattern MVC).
- Implementação do design pattern DAO.
- Arquitetura Web, Cliente/Servidor.



INSTITUTO FEDERAL
PARANÁ

Avaliação:

Proposta:

Avaliação Teórica:

Avaliação Prática:

Trabalhos:



INSTITUTO FEDERAL
PARANÁ

REDAÇÃO:

Elabore uma redação que contemple as seguintes questões:

- 1- Identificação:
- 2- Expectativa com a disciplina:
- 3- Possíveis dificuldades nos conteúdos de WEB I.
- 4- Perspectivas profissionais atuais e futuras.



INSTITUTO FEDERAL
PARANÁ

- Client-Side? (front-end)

- A aplicação roda diretamente de seu computador (browser).
- Como assim?

O Lado do cliente dá a resposta na hora pra alguma interação que é feita no website.

Ex: A maioria dos websites tem um formulário pra ser preenchido e dentro desse formulário é preciso ter validações pra ver se o campo não ficou em branco, ou se foi digitado caracteres proibidos e etc...



INSTITUTO FEDERAL
PARANÁ

A maioria dessas validações é feita em Javascript(client-side), ou seja, assim que o usuário interage com o formulário ele já responde de imediato, seja por um botão, ou por um comportamento do usuário, como mudança de foco numa caixa de texto ou até mesmo enquanto o usuário está digitando algo. O navegador não precisa buscar o código no servidor, pois este já se encontra no computador do usuário.



INSTITUTO FEDERAL
PARANÁ

Linguagens Client-Side:





INSTITUTO FEDERAL
PARANÁ

The process of client side scripting





INSTITUTO FEDERAL
PARANÁ

Vantagens:

- Economia de *bandwidth* (Largura de Banda).
- **AJAX** (Asynchronous Javascript And XML), O que o AJAX faz é o processamento externo (server-side) parecendo ser interno (client-side). O usuário não percebe que houve um novo carregamento de página, pois ele busca informações no servidor e mostra rapidamente em um local específico da página através do JavaScript. (Desempenho)



Desvantagens:

- Conhecimento do código fonte;
- Manipular o código fonte;
- A principal desvantagem do JavaScript atualmente é que o usuário pode desativá-lo em seu navegador. Se a sua aplicação basear-se exclusivamente em JavaScript, nesse caso, ela simplesmente não vai funcionar.
- Browsers não interpretar igualmente alguns scripts.



INSTITUTO FEDERAL
PARANÁ

Visão páginas HTML:

Páginas de Internet (Revista ou Jornal) **Páginas Estáticas**

Exemplo: Desenvolver site para uma editora, afim de, publicar seus 670 títulos:

- Utilizando somente HTML, ficaria impossível, pois para cada título de livro uma nova página HTML seria criada. Outro problema seria a manutenção, como em alguns casos (alteração de preço, cancelamento do produto, inclusão de novos itens).



INSTITUTO FEDERAL
PARANÁ

SOLUÇÃO:

- Criar uma página modelo e preenchê-la com informações vindas de um banco de dados, sempre que um usuário solicitar alguma informação.

PÁGINA DINÂMICA

Página montadas em um servidor web e enviadas ao usuário no momento em que ela fosse solicitada.

- Para isso, faz-se necessário o uso de SCRIPTS.



INSTITUTO FEDERAL
PARANÁ

Scripts:

- Pequenos programas que transformam páginas estáticas em páginas dinâmicas.
- São escritos utilizando linguagens de programação adequada como: PHP, VB.NET, C#, JAVA, javascript etc.



INSTITUTO FEDERAL
PARANÁ

Existem basicamente 2 maneiras de executar script:

1- Os scripts são enviados pelo servidor da web com os códigos HTML para o navegador do usuário, cabendo ao navegador interpretar esses scripts e executá-los. (client-side)

2- Os scripts são interpretados e executados pelo próprio servidor da web. O resultado desse processamento é inserido na página e enviado ao navegador do usuário em formato HTML. (server-side)



INSTITUTO FEDERAL
PARANÁ

- Server-side? (back-end)
 - São linguagens que o SERVIDOR (lado do servidor) entende, ou seja, aplicações que rodam no servidor. Isso quer dizer que você vai escrever um código onde o servidor vai processá-lo e então vai mandar para o seu navegador a resposta.

Por ex: se criarmos um script em linguagem back-end (PHP, Asp, Java, Ruby, etc) que apenas calcula a soma de $1 + 1$, será o **SERVIDOR** (ou back) que calculará este resultado.

Se fizermos esse cálculo em alguma linguagem front-end, como o JavaScript, quem calculará essa conta é o **BROWSER** do usuário. Por isso o termo client (ou front).



INSTITUTO FEDERAL
PARANÁ

Linguagens Server-Side:

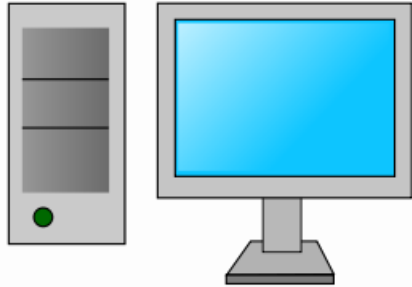




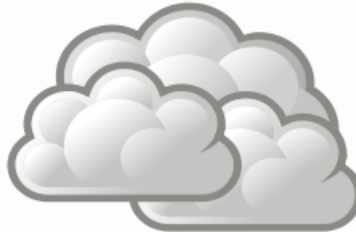
INSTITUTO FEDERAL
PARANÁ

Local Computer

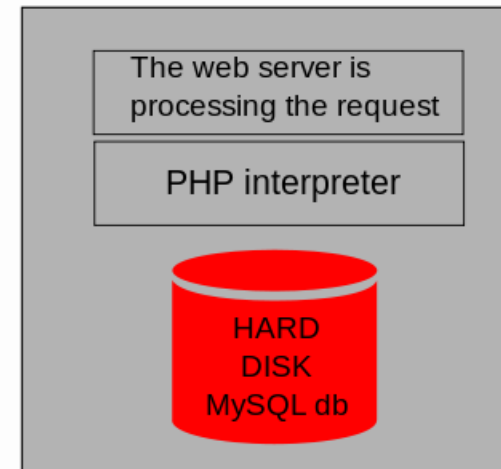
<http://www.example.com/login.php>



Internet



Web server



page in html



INSTITUTO FEDERAL
PARANÁ

Servidores Web:

- Existem para **servir** páginas web ou outros serviços, conforme sejam requisitadas por um utilizador num computador ligado à internet. É esta a função mais básica dos Servidores. Eles ficam ligados na internet e aguardam. Podem ser encontrados através de um url ou IP e quando são contactados respondem (servem) uma página web, um webservice, ou outro tipo de conteúdo, como por exemplo xml.
- São na realidade **computadores**, mas em vez de servirem para **um utilizador**, a sua função é outra. Com muita frequência nem precisam de um monitor e apenas servem para **guardar informação** (em bases de dados) e **executar procedimentos** (programação) que interagem com os utilizadores que os contactam.



INSTITUTO FEDERAL
PARANÁ



No caso das **páginas web**:

- A comunicação estabelecida com os servidores é **interpretada** pelos Browsers. São eles que contactam o servidor. Quando nós introduzimos um endereço web (URL) na barra de endereços de nosso browser (Internet Explorer, Chrome, Firefox, etc), o que o browser faz é enviar uma solicitação ao servidor que gere aquele URL. De seguida, os Servidores servem o conteúdo que nosso browser põe legível e bonitinho para vermos.



INSTITUTO FEDERAL
PARANÁ

Tipos de servidores: Existem vários tipos de servidores e as vantagens ou desvantagens de cada um dependem das funcionalidades a que se destinam.





INSTITUTO FEDERAL
PARANÁ

Pacotes de instalação:

W.A.M.P significa: **W**indows, **A**pache, **M**ysql/**M**ariaDB, **P**HP

X.A.M.P.P significa: **X** (qualquer sistema), **A**pache, **M**ysql/**M**ariaDB, **P**HP e **P**erl

EasyPHP: Tem os seguintes pacotes PHP, Apache, MySQL, Nginx (alternativa ao apache, pode alternar entre ambos), PhpMyAdmin, PostgreSQL, MongoDB, Python, Ruby e roda apenas em Windows, tem pacotes para versões de processadores x86 e x64.



INSTITUTO FEDERAL
PARANÁ

Editores de texto:

- NotePad;
- Sublime;
- Notepad++;
- EclipsePHP;
- Zend Studio;
- Etc...



INSTITUTO FEDERAL
PARANÁ

PHP

Personal Home Page



INSTITUTO FEDERAL
PARANÁ

O que é o PHP?

- A linguagem de programação PHP, foi criada no outono de 1994 por Rasmus Lerdorf.
- No início era formada por um conjunto de scripts voltados à criação de páginas dinâmicas que Rasmus utilizava para monitorar o acesso ao seu currículo na internet. À medida que essa ferramenta foi crescendo em funcionalidade, Rasmus teve de escrever uma implementação em C, a qual permitia às pessoas desenvolverem de forma muito simples suas aplicações para web.



INSTITUTO FEDERAL
PARANÁ

- Surge PHP/FI (Personal Home Pages/Forms Interpreter), disponibilizada seu código na web, em 1995, para compartilhar com outras pessoas, bem como receber ajuda e correção de bugs.
- Em novembro de 1997 foi lançada a segunda versão do PHP. Naquele momento, aproximadamente 1% da internet já utilizavam PHP.
- Andi Gutmans e Zeev Suraski (estudantes) cooperaram com Rasmus para aprimorar o PHP.
- Surge o PHP 3 (junho de 1998), características – estabilidade, possibilidade de conexão com vários BD, novos protocolos, sintaxe consistente, suporte a OO e uma nova API (possibilitava criar novos módulos).



- No final de 1998, PHP presente em 10% dos domínios da Internet. PHP(Hypertext Preprocessor).
- PHP 4, é lançado em maio de 2000, trazendo muitas melhorias e recursos novos, como seções, suporte a diversos servidores web, abstração da API. PHP presente em cerca de 20% dos domínios da internet.
- PHP 5, lançada em julho de 2004, teve por objetivo aprimorar o suporte a orientação a objetos.
- Versões PHP 5.3, 5.4, 5.5 (Resolver o problema do Unicode(textos de qualquer tipo de escrita)).
- PHP 7, lançada em agosto de 2015, versão de correção de bugs. (PESQUISAR).



INSTITUTO FEDERAL
PARANÁ

Extensão de Arquivos:

Extensão:

.php	Arquivo PHP contendo um programa.
.class.php	Arquivo PHP contendo uma classe.
.inc.php	Arquivo PHP a ser incluído, pode incluir constantes ou config.

Pode também:

.php3	Arquivo PHP contendo um programa PHP versão 3.
.php4	Arquivo PHP contendo um programa PHP versão 4.
.phtml	Arquivo PHP contendo um programa PHP e HTML.



INSTITUTO FEDERAL
PARANÁ

Delimitadores de código:

```
<?php  
// código;  
// código;  
// código;  
?>
```



INSTITUTO FEDERAL
PARANÁ

Comentários:

Para comentar uma única linha:

```
// echo "a";  
# echo "a";
```

Para comentar muitas linhas:

```
/* echo "a";  
   echo "b"; */
```



INSTITUTO FEDERAL
PARANÁ

Comandos de saída (output):

echo

É um comando que imprime uma ou mais variáveis no console.

EX:

echo 'a', 'b', 'c';

Resultado:

abc



INSTITUTO FEDERAL
PARANÁ

Comandos de saída (output):

print

É uma função que imprime uma string no console.

EX:

```
print ('abc');
```

Resultado:

abc



INSTITUTO FEDERAL
PARANÁ

Comandos de saída (output):

var_dump

Imprime o conteúdo de uma variável de forma explanativa, muito comum para se realizar **debug**. Se o parâmetro for um objeto, ele imprimirá todos os seus atributos; se for um array de várias dimensões, imprimirá todas elas, com seus respectivos conteúdos e tipos de dados.

EX:

```
$vetor = array('Palio', 'Gol', 'Fiesta', 'Corsa');  
var_dump($vetor);
```

Resultado:

```
array(4) { [0]=> string(5) "Palio" [1]=> string(3) "Gol" [2]=> string(6) "Fiesta"  
[3]=> string(5) "Corsa" }
```



INSTITUTO FEDERAL
PARANÁ

Comandos de saída (output):

`print_r`

Imprime o conteúdo de uma variável de forma explanativa, assim como a **`var_dump()`**, mas em um formato mais legível para o programador, com os conteúdos alinhados e suprimindo os tipos de dados.

EX:

```
$vetor = array('Palio', 'Gol', 'Fiesta', 'Corsa');  
print_r($vetor);
```

Resultado:

```
Array ( [0] => Palio [1] => Gol [2] => Fiesta [3] => Corsa )
```



INSTITUTO FEDERAL
PARANÁ

Variáveis:

São identificadores utilizados para representar valores mutáveis e voláteis, que só existem durante a execução do programa. Elas são armazenadas na memória RAM e seu conteúdo é destruído após a execução do programa.

Para criar uma variável em PHP, precisa-se atribuir um nome de identificação, sempre precedido pelo caractere cifrão (\$).



INSTITUTO FEDERAL
PARANÁ

Exemplo:

```
<?php
    $nome = "João";
    $sobrenome = "da Silva";
    echo "$sobrenome, $nome";
?>
```

Resultado:

da Silva, João



INSTITUTO FEDERAL
PARANÁ

DICAS:

- Nunca inicie a nomenclatura de variáveis com números;
- Nunca utilize espaços em branco no meio do identificador da variável;
- Nunca utilize caracteres especiais (! @ # % ^ & * / |[] { }) na nomenclatura das variáveis;
- Evite criar variáveis com mais de 15 caracteres em virtude da clareza do código fonte;
- Nomes de variáveis devem ser significativos e transmitir a ideia de seu conteúdo dentro do contexto no qual a variável está inserida.
- Utilize preferencialmente palavras em minúsculo (separadas pelo caractere “_”) ou somente as primeira letras em maiúsculo quando da ocorrência de mais palavras.



INSTITUTO FEDERAL
PARANÁ

Exemplo:

```
<?php
    $codigo_cliente
    $codigoCliente
?>
```

O PHP é *case sensitive*, ou seja, é sensível a letras maiúsculas e minúsculas. Tome cuidado ao declarar variáveis e nomes de funções. Por exemplo, a variável \$codigo é tratada de forma totalmente diferente da variável \$Codigo.



INSTITUTO FEDERAL
PARANÁ

Variáveis variantes (*variable variables*)

Em alguns casos, precisa-se ter em nosso código-fonte nomes de variáveis que podem mudar de acordo com determinada situação. Neste caso não somente o conteúdo da variável é mutável, mas também seu nome.

Sempre que utilizar dois (\$) precedendo o nome de uma variável, o PHP irá referenciar uma variável representada pelo conteúdo da primeira.



INSTITUTO FEDERAL
PARANÁ

Variáveis variantes (*variable variables*)

```
<?php
// define o nome da variável
$variavel = 'nome';
// cria variável identificada pelo conteúdo de $variavel
$$variavel = 'maria';
// exibe variável $nome na tela
echo $nome;
?>
```




INSTITUTO FEDERAL
PARANÁ

Quando uma variável é atribuída a outra, sempre é criada uma nova área de armazenamento na memória. Exemplificando:

```
<?php
    $a = 5;
    $b = $a;
    $b = 10;
    echo $a;    // resultado = 5
    echo $b;    // resultado = 10
?>
```



INSTITUTO FEDERAL
PARANÁ

Referência entre variáveis:

Ocorre quando precisamos que duas variáveis apontem para a mesma região de memória, a atribuição deve ser precedida pelo operador &. Assim, qualquer alteração em qualquer uma das variáveis reflete na outra.

```
<?php
    $a = 5;
    $b = &$a;
    $b = 10;
    echo $a; // resultado = 10;
    echo $b; // resultado = 10;
?>
```



INSTITUTO FEDERAL
PARANÁ

Tipos de variáveis



Booleano

- Expressa um valor lógico que pode ser **verdadeiro** ou **falso** (TRUE ou FALSE).
- Ex:

```
<?php
```

```
//declara variável com valor TRUE
```

```
$exibir_nome = TRUE;
```

```
// testa se $exibir_nome é TRUE
```

```
if($exibir_nome)
```

```
{
```

```
    echo 'Aqui passou por ser TRUE';
```

```
}
```

```
?>
```



INSTITUTO FEDERAL
PARANÁ

Booleano

- Ex:

```
<?php
```

```
// declara variável numérica
```

```
$umidade = 91;
```

```
// testa se é maior que 90. Retorna um boolean
```

```
$vai chover = ($umidade > 90);
```

```
// testa se $vai_chover é verdadeiro
```

```
if($vai_chover)
```

```
{
```

```
    echo 'Está chovendo';
```

```
}
```

```
?>
```



INSTITUTO FEDERAL
PARANÁ

Também são considerados valores falsos em comparações booleanas:

- Inteiro 0
- Ponto flutuante 0.0
- Uma string vazia "" ou "0"
- Um array vazio
- Um objeto sem elementos
- Tipo NULL



INSTITUTO FEDERAL
PARANÁ

Numérico:

Números podem ser especificados em notação decimal (base 10), hexadecimal (base 16) ou octal (base 8), opcionalmente precedidos de sinal (- ou +).



INSTITUTO FEDERAL
PARANÁ

```
<?php
```

```
// número decimal
```

```
$a = 1234;
```

```
// número negativo
```

```
$a = -123;
```

```
// número octal (equivale a 83 em decimal)
```

```
$a = 0123;
```

```
// número hexadecimal (equivale a 26 em decimal)
```

```
$a = 0x1A;
```

```
// ponto flutuante
```

```
$a = 1.234;
```

```
// notação científica
```

```
$a = 4e23;
```

```
?>
```



INSTITUTO FEDERAL
PARANÁ

String:

É uma cadeia de caracteres alfanuméricos. Para declará-la pode-se utilizar aspas simples ' ' ou aspas duplas " ".

```
<?php  
    $variavel = 'Isto é um teste';  
    $variavel = "Isto é um teste";  
?>
```



INSTITUTO FEDERAL
PARANÁ

Array:

Lista de valores armazenados na memória, os quais podem ser de tipos diferentes (números, strings, objetos) e podem ser acessados a qualquer momento, pois cada valor é relacionado a uma chave. Um array pode crescer dinamicamente com a adição de novos itens.

```
<?php
    $carros = array('Palio', 'Corsa', 'Fox');
    echo $carros[1];
?>
```

Resultado = 'Corsa'



INSTITUTO FEDERAL
PARANÁ

Tipo objeto:

Um objeto é uma entidade com um determinado comportamento definido por seus métodos (ações) e propriedades (dados “atributos”). Para criar um objeto deve-se utilizar o operador *new*.



INSTITUTO FEDERAL
PARANÁ

Tipo objeto:

```
<?php
class Computador {
    var $cpu;
    function ligar() {
        echo "Ligando computador a {$this->cpu}...";
    }
}
$obj = new Computador;
$obj -> cpu = "2.3Ghz";
$obj -> ligar();
?>
```

Resultado = Ligando computador a 2.3Ghz



INSTITUTO FEDERAL
PARANÁ

Tipo recurso:

Recurso (*resource*) é uma variável especial que mantém uma referência de recurso externo. Recursos são criados e utilizados por funções especiais, como uma conexão ao banco de dados. Um exemplo é a função *mysql_connect()*, que, ao conectar-se ao banco de dados, retorna uma variável de referência do tipo recurso.

```
resource mysql_connect(...)
```



INSTITUTO FEDERAL
PARANÁ

Constantes:

Uma constante é um valor que não sofre modificações durante a execução do programa. Ela é representada por um identificador, assim como as variáveis, com a exceção de que só pode conter valores escalares (boolean, inteiro, ponto flutuante e string). Um valor escalar não pode ser composto de outros valores, como vetores ou objetos. As regras de nomenclatura de constantes seguem as mesmas regras das variáveis, com a exceção de que as constantes não são precedidas pelo sinal de cifrão (\$) e geralmente utilizam-se nomes em maiúsculo.



INSTITUTO FEDERAL
PARANÁ

Constantes:

Ex:

```
<?php  
define("MAXIMO_CLIENTES", 100);  
echo MAXIMO_CLIENTES;  
?>
```

Resultado: 100



INSTITUTO FEDERAL
PARANÁ

Operadores



INSTITUTO FEDERAL
PARANÁ

Atribuição:

Um operador de atribuição é utilizado para definir uma variável atribuindo-lhe um valor. O operador básico de atribuição é =.

```
<?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;  
?>
```



INSTITUTO FEDERAL
PARANÁ

Atribuição:

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



INSTITUTO FEDERAL
PARANÁ

Aritméticos:

São utilizados para realizar cálculos matemáticos:

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



INSTITUTO FEDERAL
PARANÁ

Aritméticos:

Cálculos complexos, procure utilizar parênteses, sempre observando as prioridades aritméticas.

```
<?php  
$a = 2;  
$b = 4;  
echo $a+3*4+5*$b;      // resultado = 34  
echo ($a + 3)*4+(5*$b); // resultado = 40  
?>
```



INSTITUTO FEDERAL
PARANÁ

Aritméticos:

O PHP realiza automaticamente a conversão de tipos em operações:

```
<?php  
$a = '10';  
//soma + 5;  
echo $a + 5;  
?>
```

Resultado: 15



INSTITUTO FEDERAL
PARANÁ

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.



INSTITUTO FEDERAL
PARANÁ

Testes Lógicos:

```
<?php  
If($a = 5)  
{  
    echo 'essa operação atribui 5 à variável $a e retorna verdadeiro';  
}  
?>
```



INSTITUTO FEDERAL
PARANÁ

Testes Lógicos (Comparação == e !=):

```
<?php  
$a = 1234;  
$b = '1234';
```

```
if($a == $b)  
{  
    echo '$a e $b são iguais';  
} else if ($a != $b) {  
    echo '$a e $b são diferentes';  
}  
?>
```

resposta: \$a e \$b são iguais



INSTITUTO FEDERAL
PARANÁ

Testes Lógicos (Comparação === e !==):

```
<?php
```

```
$c = 1234;
```

```
$d = '1234';
```

```
if($c === $d)
```

```
{
```

```
    echo '$c e $d são iguais e do mesmo tipo';
```

```
} else if ($a !== $b) {
```

```
    echo '$a e $b são de tipos diferentes';
```

```
}
```

```
?>
```

resposta: ?



INSTITUTO FEDERAL
PARANÁ

Testes Lógicos (0 é considerado FALSO em comparações):

```
<?php
$e = 0;
// zero sempre é igual à FALSE
If($e == FALSE){
    echo '$e é falso';
}
//testa se $e é do tipo FALSE
```

resposta: ?

```
if ($e === FALSE) {
    echo '$e é do tipo false';
}
if ($e === 0) {
    echo '$e é zero';
}
?>
```



INSTITUTO FEDERAL
PARANÁ

Testes Lógicos (0 é considerado FALSO em comparações):

```
<?php
$e = 0;
// zero sempre é igual à FALSE
If($e == FALSE){
    echo '$e é falso';
}
//testa se $e é do tipo FALSE
```

resposta: e é falso
e é zero

```
if ($e === FALSE) {
    echo '$e é do tipo false';
}
if ($e === 0) {
    echo '$e é zero';
}
?>
```



INSTITUTO FEDERAL
PARANÁ

Lógicos:

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

OBS: or e and têm precedência menor que && ou ||.



INSTITUTO FEDERAL
PARANÁ

Lógicos:

```
<?php  
$vai_chover = TRUE;  
$esta_frio = TRUE;  
  
If($vai_chover and $esta_frio)  
{  
    echo "Não vou sair de casa";  
}  
?>
```

Resposta: ?



INSTITUTO FEDERAL
PARANÁ

Lógicos:

```
<?php
$vai_chover = FALSE;
$esta_frio = TRUE;

If($vai_chover xor $esta_frio)
{
    echo "Vou pensar duas vezes antes de sair";
}
?>
```

Resposta: ?



INSTITUTO FEDERAL
PARANÁ

Lógicos:

```
<?php
$vai_chover = FALSE;
$esta_frio = TRUE;

If($vai_chover xor $esta_frio)
{
    echo "Vou pensar duas vezes antes de sair";
}
?>
```

Resposta: ?



INSTITUTO FEDERAL
PARANÁ

Estruturas de controle



INSTITUTO FEDERAL
PARANÁ

IF:

Estrutura de controle que introduz um desvio condicional, ou seja, um desvio na execução natural do programa. Caso a condição dada pela expressão seja satisfeita, então serão executadas as instruções do bloco de comandos. Caso a condição não seja satisfeita, o bloco de comandos será ignorado. O comando IF pode ser lido como:

“SE (expressão) ENTÃO { *comandos...* }”.



INSTITUTO FEDERAL
PARANÁ

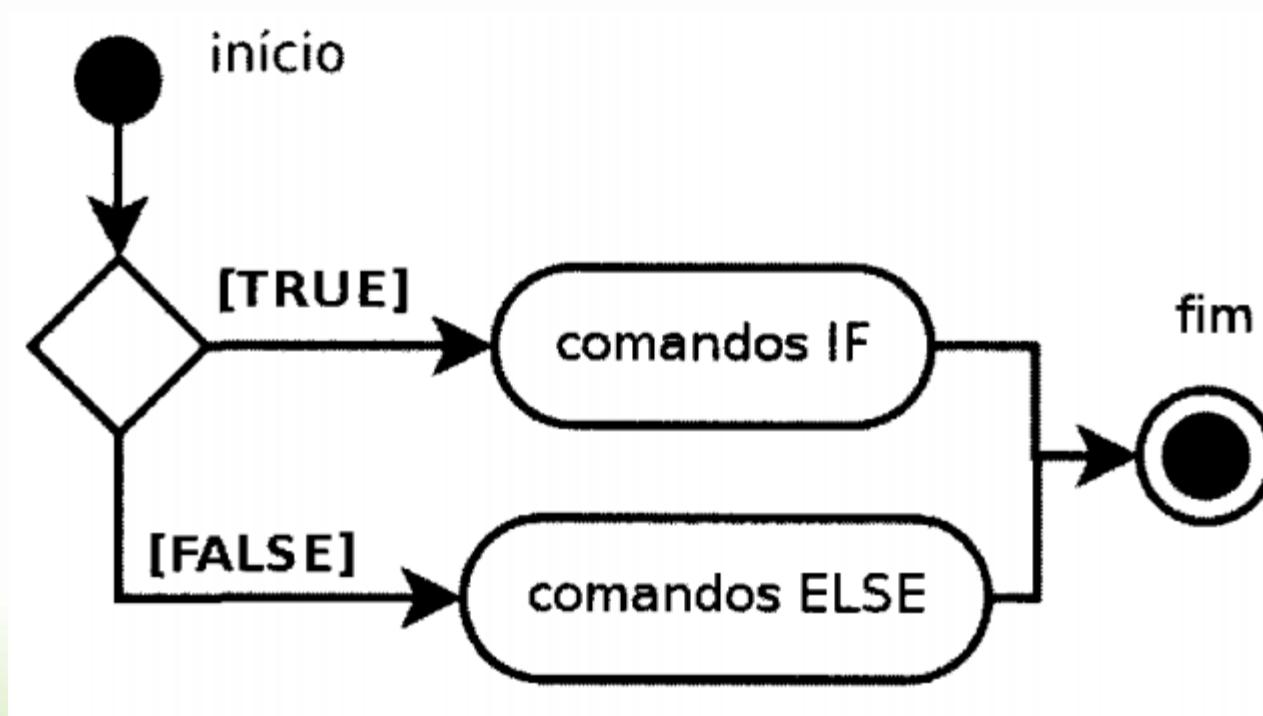
ELSE:

Utilizado para indicar um novo bloco de comandos delimitados por { }, caso a condição do IF não seja satisfeita. Pode ser lido como “caso contrário”. A utilização do ELSE é opcional.



INSTITUTO FEDERAL
PARANÁ

Fluxograma:





INSTITUTO FEDERAL
PARANÁ

Sintáxe no PHP:

```
if (expressão)
{
    comandos se expressão é verdadeira;
}
else
{
    comandos se expressão é falsa;
}
```



INSTITUTO FEDERAL
PARANÁ

Exemplo:

```
<?php
$a = 1;
if ($a == 5)
{
    echo "é igual";
}
else
{
    echo "não é igual";
}
?>
```



INSTITUTO FEDERAL
PARANÁ

Exemplo:

```
<?php
$a = 'conteúdo';
if ($a)
{
    echo '$a tem conteúdo';
}
if ($b)
{
    echo "$b tem conteúdo";
}
?>
```



INSTITUTO FEDERAL
PARANÁ

Exemplo: IF Encadeado

```
<?php
$salario = 1020;    // R$
$tempo_servico = 12; // meses
$tem_reclamacoes = false; // booleano
if ($salario > 1000)
{
    if ($tempo_servico >= 12)
    {
        if ($tem_reclamacoes != true)
        {
            echo 'parabéns, você foi promovido';
        }
    }
}
?>
```



INSTITUTO FEDERAL
PARANÁ

Exemplo: Ou pode-se:

```
<?php
$salario = 1020;    // R$
$tempo_servico = 12; // meses
$tem_reclamacoes = false; // booleano

if (($salario > 1000) and ($tempo_servico >= 12) and ($tem_reclamacoes != true )){
    echo 'parabéns, você foi promovido';
}
if (($salario > 1000) && ($tempo_servico >= 12) && ($tem_reclamacoes != true )){
    echo 'parabéns, você foi promovido';
}
?>
```



INSTITUTO FEDERAL
PARANÁ

Exemplo:

```
<?php
$valor_venda = 120;

if ($valor_venda > 100)
{
    $resultado = 'muito caro';
    echo $resultado;
}
else
    $resultado = 'pode comprar';
    echo resultado;
}
?>
```



INSTITUTO FEDERAL
PARANÁ

Exemplo: Ou pode-se utilizar de ternário

```
<?php
```

```
$valor_venda = 90;
```

```
$resultado = ($valor_venda > 100) ? 'muito caro' : 'pode comprar';
```

```
echo $resultado;
```

```
?>
```




While

- Estabelece um laço de repetição, ou seja, o bloco de comandos será executado repetitivamente enquanto a condição de entrada dada pela expressão for verdadeira.

Interpretação: “ENQUANTO (expressão) FAÇA {comandos...}..”

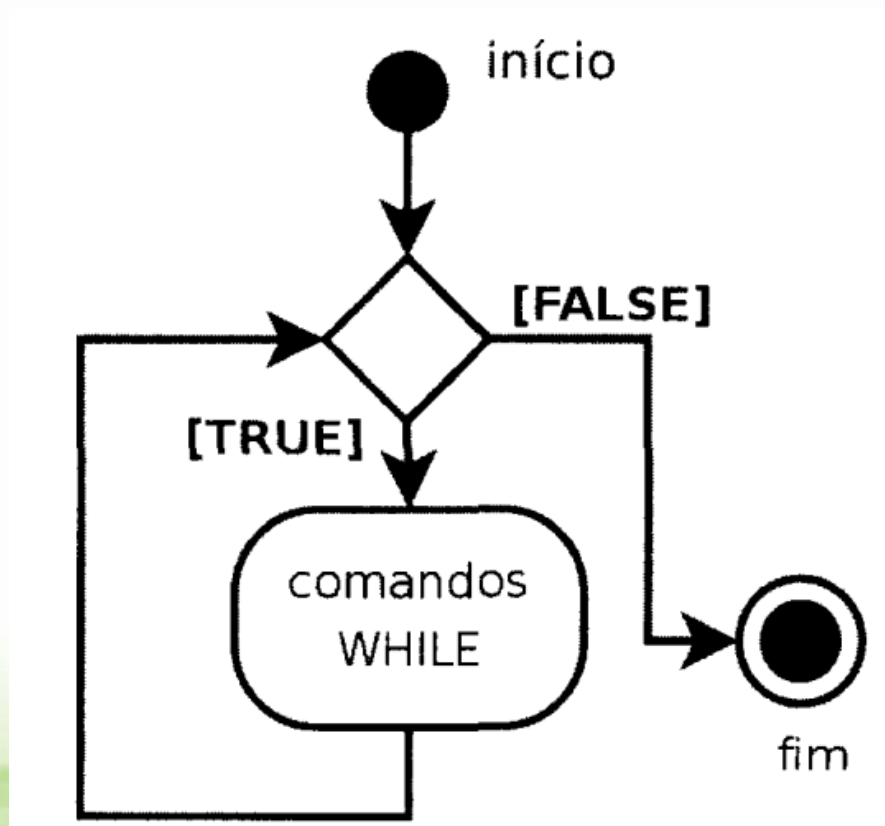
SINTAXE:

```
while (expressão)
{
    comandos;
}
```



INSTITUTO FEDERAL
PARANÁ

Fluxograma:





INSTITUTO FEDERAL
PARANÁ

Exemplo:

```
<?php  
$a = 1;  
while($a < 5)  
{  
    print $a;  
    $a ++;  
}
```

Resultado: 1234



FOR:

- Estabelece um laço de repetição baseado em um contador. O FOR é controlado por um bloco de três comandos que estabelecem uma contagem, ou seja, o bloco de comandos será executado um certo numero de vezes.

SINTAXE:

```
for(expr1; expr2; expr3)
{
    comandos...
}
```

expr1 = Valor inicial da variável contadora.

expr2 = Condição de execução. Enquanto for TRUE, o bloco de comandos será executado.

expr3 = Valor a ser incrementado após cada execução.



INSTITUTO FEDERAL
PARANÁ

Exemplo:

```
<?php
    for($i = 1; $i <= 10; $i++)
    {
        $j = 2;
        $resp = $j * $i;
        print "$resp <br>";
    }
?>
```

Resultado: ?



INSTITUTO FEDERAL
PARANÁ

Exemplo: Contadores permite variáveis de apenas uma letra...

```
<?php
```

```
for ($i = 1; $i <= 10; $i++)
```

```
{
```

```
    print "<br>";
```

```
    for ($j = 1; $j <= 10; $j++)
```

```
    {
```

```
        $resp = $i * $j;
```

```
        print $i."X".$j."= $resp <br>";
```

```
    }
```

```
}
```

```
?>
```

resposta: ?



INSTITUTO FEDERAL
PARANÁ

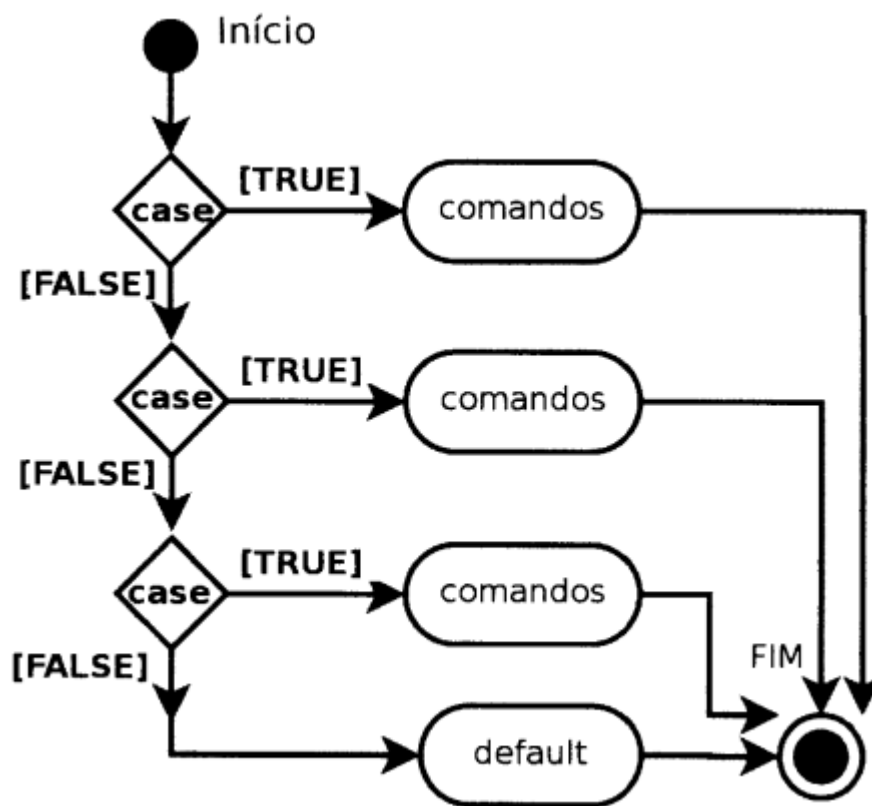
SWITCH:

- Estrutura que simula uma bateria de testes sobre uma variável. Similar a uma série de comandos IF sobre a mesma expressão. Frequentemente, é necessário comparar a mesma variável com valores diferentes e executar uma ação específica em cada um destes casos.



INSTITUTO FEDERAL
PARANÁ

Fluxograma:





INSTITUTO FEDERAL
PARANÁ

Sintaxe:

```
switch ($expressão)
{
    case "valor 1":
        // comandos
        break;
    case "valor 2":
        // comandos
        break;
    case "valor n":
        // comandos
        break;
    default:
        // comandos
}
```



INSTITUTO FEDERAL
PARANÁ

Exemplos:

```
<?php
```

```
$i = 1;
```

```
Switch ($i)
```

```
{
```

```
    case 0:
```

```
        print "i é igual a 0";
```

```
        break;
```

```
    case 1:
```

```
        print "i é igual a 1";
```

```
        break;
```

```
    case 2:
```

```
        print "i é igual a 2";
```

```
        break;
```

```
    default:
```

```
        print "i não é igual a 0, 1 ou 2";
```

```
}
```

```
?>
```



INSTITUTO FEDERAL
PARANÁ

FOREACH:

- Laço de repetição para iterações em *array* ou matrizes. É um FOR simplificado que decompõe um vetor ou matriz em cada um de seus elementos por meio de sua cláusula AS.

SINTAXE:

```
foreach ($array as $valor)
{
    instruções
}
```



INSTITUTO FEDERAL
PARANÁ

Exemplo:

```
<?php
$fruta = array("maça", "laranja", "pêra", "banana");
Foreach ($fruta as $valor)
{
    print "$valor -";
}
?>
```



INSTITUTO FEDERAL
PARANÁ

Requisição de arquivos:

- Em linguagem de script como o PHP, frequentemente precisamos incluir dentro do programa outros arquivos, sejam esses, funções, constantes, configurações, ou mesmo carregar um arquivo contendo a definição de uma classe:
- Sintaxe:

Include <arquivo>

A instrução **include()** inclui e avalia o arquivo informado.



INSTITUTO FEDERAL
PARANÁ

Exemplo:

bibliteca.php

```
<?php
```

```
/*
```

- Função quadrado
- Retorna o quadrado de um número

```
*/
```

```
function quadrado($numero)
```

```
{
```

```
    return $numero * $numero;
```

```
}
```

```
?>
```




INSTITUTO FEDERAL
PARANÁ

Exemplo:

teste.php

```
<?php
```

```
// Carregar arquivo com a função necessária  
include 'biblioteca.php';
```

```
// Imprime o quadrado do número 4
```

```
echo quadrado(4);
```

```
?>
```



INSTITUTO FEDERAL
PARANÁ

Exercício:

- 1) Assinale a alternativa que representa tipos de variáveis escalares em PHP:
 - a) Boolean, NULL e Array
 - b) Integer, Float e String
 - c) Float, String e Object
 - d) String, Boolean e Array
 - e) Array, Recurso e NULL



INSTITUTO FEDERAL
PARANÁ

Exercício:

2) De acordo com o trecho de código escrito na linguagem PHP:

```
$a = array("a", "b", "c", "d");  
$a[] = "e";
```

A variável \$a conterá:

- a) e
- b) e, a, b, c, d
- c) não é possível atribuir um valor a um array desta forma
- d) a, b, c, d, e
- e) o array receberá um valor nulo



INSTITUTO FEDERAL
PARANÁ

Exercício:

3) De acordo com o trecho de código escrito na linguagem PHP:

```
$b = 0; $a = $b + "a";
```

O conteúdo da variável \$a será:

- a) ab
- b) 0 + a
- c) 0
- d) 1
- e) nenhuma das alternativas anteriores



INSTITUTO FEDERAL
PARANÁ

Exercício:

4) De acordo com o trecho de código escrito na linguagem PHP:

```
$c = "3 navios" + "10 submarinos";
```

```
echo $c;
```

O conteúdo da variável \$c será:

- a) 13 navios 10 submarinos;
- b) NULL
- c) 3
- d) 13
- e) nenhuma das alternativas anteriores



INSTITUTO FEDERAL
PARANÁ

Exercício:

5) Assinale a alternativa CORRETA para o trecho de código escrito na linguagem PHP, que exibirá a saída abaixo:

Ela me disse: "Gosto mais do Bob's que do Mac".
Então fomos para o Bob's.

- a) `echo 'Ela me disse: 'Gosto mais do Bob\'s que do Mac'. Então fomos para o Bob's.';`
- b) `echo 'Ela me disse: '."Gosto mais do Bob's que do Mac.". 'nPorém prefiro almoçar em casa.';`
- c) `echo "Ela me disse: \"Gosto mais do Bob's que do Mac\". Então fomos para o Bob's.";`
- d) `echo "Ela me disse: \"Gosto mais do Bob\'s que do Mac\".nEntão fomos para o Bob's.";`
- e) `echo "Ela me disse: \"Gosto mais do Bob's que do Mac\".nEntão fomos para o Bob's.";`



INSTITUTO FEDERAL
PARANÁ

Exercício:

6) De acordo com o trecho de código escrito na linguagem em PHP:

```
$i = 0;  
while(1) {  
    if($i == 10)  
        break;  
    echo ++$i . "\n";  
}
```

O resultado deste código irá retornar:

- a) Intervalo de 1 a 10
- b) Intervalo de 0 a 9
- c) Intervalo de 0 a 10
- d) 10
- e) nenhuma das alternativas anteriores