

Introdução ao PHP e SQL

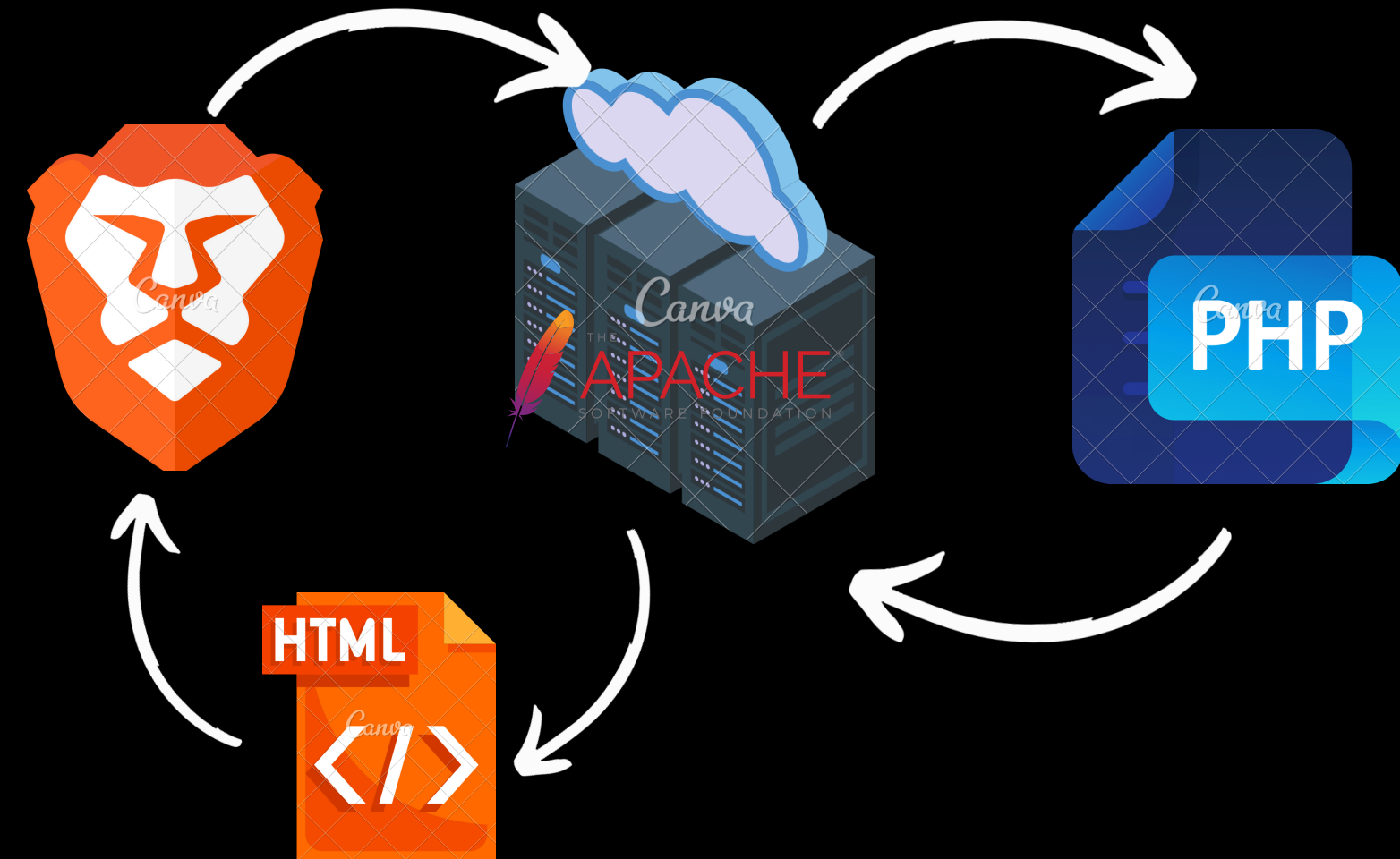
IFBA–Jacobina

Alunos: João Vitor Silvestre, Xanderson Silva

Turma: 1814

Semana Nacional de Ciência e Tecnologia (SNCT) 2024

Para que serve/ Como funciona



XAMPP

Significado da sigla

X - Sistema Operacional

A - Apache - Servidor

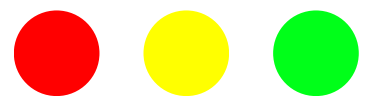
M - MySQL - Banco de dados

P - PHP - Linguagem

P - Perl - Linguagem (Opcional)

Variáveis e Constantes em PHP

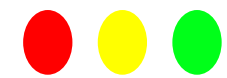
Estruturas condicionais (como if e else) controlam o fluxo do programa, executando comandos conforme condições são atendidas.



```
pc0@dev> $nome = "Maria"; //Variável  
pc0@dev>  
pc0@dev> define("PI", 3.14 ); //Constante
```

Estruturas Condicionais em PHP

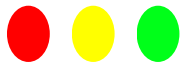
Estruturas condicionais (como if e else) controlam o fluxo do programa, executando comandos conforme condições são atendidas.



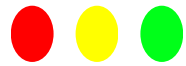
```
if ($idade >= 18) {  
    echo "Adulto";  
} else {  
    echo "Menor de idade";  
}
```

Laços de Repetição

Laços permitem repetir blocos de código até que uma condição seja satisfeita.



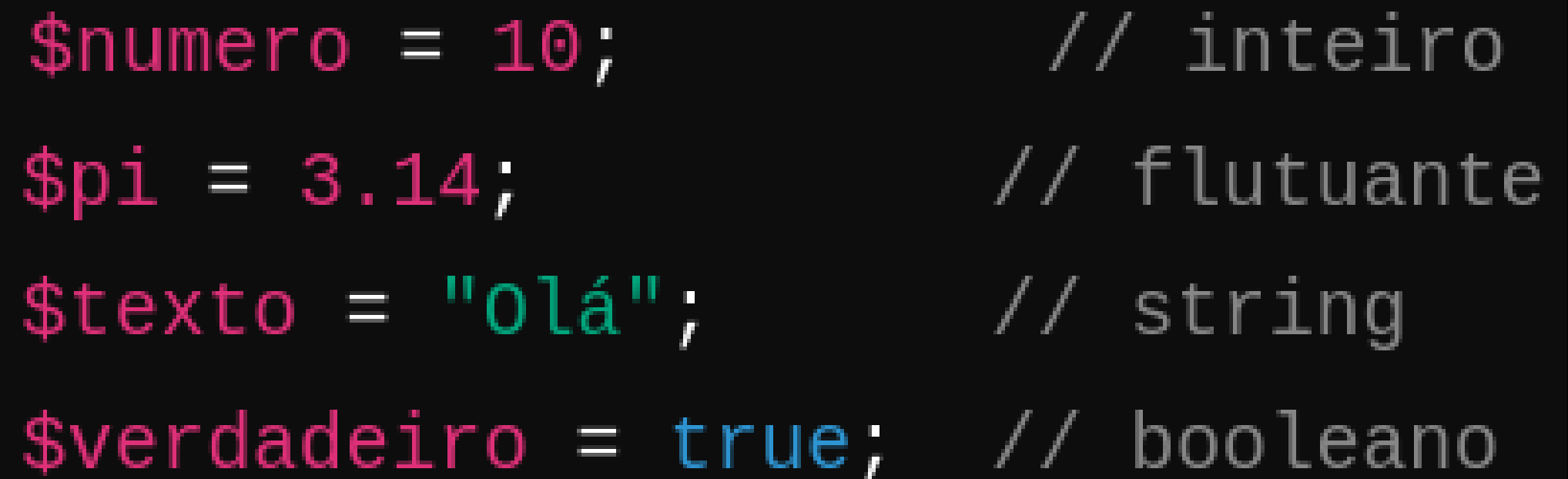
```
for ($i = 0; $i < 5; $i++) {  
    echo $i;  
}
```



```
$i = 0;  
while ($i < 5) {  
    echo $i;  
    $i++;  
}
```

Tipos Primitivos

- Inteiros
- Flutuantes
- Strings
- Booleanos



```
$numero = 10;           // inteiro  
$pi = 3.14;             // flutuante  
$texto = "olá";         // string  
$verdadeiro = true;     // booleano
```


Manipulação de Strings

- `strlen()`
- `strtoupper()`
- `strtolower()`

Operadores Aritméticas

- Soma (+)
- Subtração (-)
- Multiplicação (*)
- Divisão (/)
- Módulo (%)

Funções Aritméticas do PHP

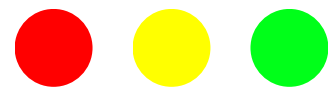
- `round()`
- `pow()`
- `sqrt()`
- `max()`
- `min()`

Funções Aritméticas do PHP

- `round()`
- `pow()`
- `sqrt()`
- `max()`
- `min()`

Superglobais no PHP

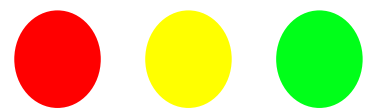
- \$_GET
- \$_POST
- \$_SESSION
- \$_COOKIE
- \$_SERVER



```
session_start();  
$_SESSION['username'] = 'Maria';  
  
setcookie("usuario", "Maria", time() + 3600); // cookie por 1 hora
```

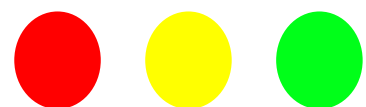
Obtendo Dados de Formulários

- Captura de dados usando `$_POST` ou `$_GET`



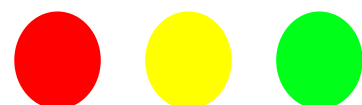
```
<?php
// Exemplo de código para capturar e processar notas enviadas via formulário
$nota1 = $_POST['nota1'];
$nota2 = $_POST['nota2'];
$media = ($nota1 + $nota2) / 2;
echo "A média das notas é: " . $media;
?>
```

Formulário retroalimentado



```
<h2>Calculadora de Média</h2>
<form method="post" action="">
  <label for="nota1">Nota 1:</label>
  <input type="number" name="nota1" id="nota1" step="0.01" required>
  <br><br>
  <label for="nota2">Nota 2:</label>
  <input type="number" name="nota2" id="nota2" step="0.01" required>
  <br><br>
  <button type="submit">Calcular Média</button>
</form>
```

Formulário retroalimentado

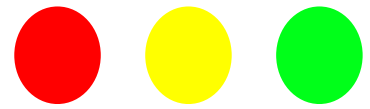


```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Obtendo as notas dos campos do formulário
    $nota1 = $_POST['nota1'];
    $nota2 = $_POST['nota2'];

    // Calculando a média
    $media = ($nota1 + $nota2) / 2;

    // Exibindo o resultado
    echo "<h3>Resultado</h3>";
    echo "<p>A média das notas é: " . number_format($media, 2, ',', '.') . "</p>"
}
?>
```


Formulário retroalimentado



```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Obtendo as notas dos campos do formulário
    $nota1 = $_POST['nota1'];
    $nota2 = $_POST['nota2'];

    // Calculando a média
    $media = ($nota1 + $nota2) / 2;

    // Exibindo o resultado
    echo "<h3>Resultado</h3>";
    echo "<p>A média das notas é: " . number_format($media, 2, ',', '.') . "</p>"
}
?>
```

SQL

Data Manipulation Language (DML)

- o CREATE – usado para criar o banco de dados ou objetos
- o DROP – usado para deletar objetos do banco de dados;
- o ALTER – usado para alterar a estrutura do banco de dados.

Data Definition Language (DDL)

- o SELECT – usado para recuperar dados do BD
- o INSERT – usado para inserir dados em uma tabela;
- o UPDATE – usado para atualizar dados existentes em uma tabela
- o DELETE – usado para apagar dados de uma tabela.

Data Manipulation Language

Criar Banco de Dados

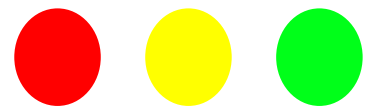
```
>>> CREATE DATABASE NomeDoBanco;
```

Excluir um Banco de Dados

```
>>> DROP DATABASE NomeDoBanco;
```

Data Manipulation Language

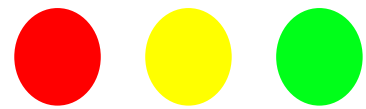
CREATE – usado para criar o banco de dados ou objetos



```
CREATE TABLE clientes (  
    id INT PRIMARY KEY,  
    nome VARCHAR(255) NOT NULL,  
    email VARCHAR(255) UNIQUE,  
    telefone VARCHAR(20),  
    endereco VARCHAR(255),  
    cidade VARCHAR(100),  
    estado VARCHAR(50),  
    cep VARCHAR(20),  
    data_nascimento DATE  
);
```

Data Manipulation Language

ALTER – usado para alterar a estrutura do banco de dados.

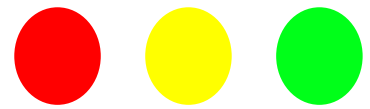


```
ALTER TABLE clientes ADD COLUMN email VARCHAR(255);
```

Data Definition Language

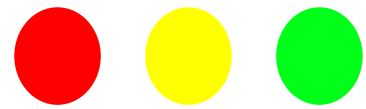
- SELECT - usado para recuperar dados do BD
- INSERT - usado para inserir dados em uma tabela;
- UPDATE - usado para atualizar dados existentes em uma tabela
- DELETE - usado para apagar dados de uma tabela.

Data Definition Language



```
INSERT INTO clientes (id, nome, email, telefone, endereco, cidade, estado, cep,
data_nascimento) VALUES
(1, 'Maria Silva', 'maria.silva@email.com', '(11) 99999-9999', 'Rua das Flores, 123',
'São Paulo', 'SP', '01234-567', '1985-03-15'),
(2, 'João Souza', 'joao.souza@email.com', '(11) 88888-8888', 'Rua dos Lírios, 456',
'Rio de Janeiro', 'RJ', '23456-789', '1990-07-22'),
(3, 'Ana Oliveira', 'ana.oliveira@email.com', '(11) 77777-7777', 'Rua das Margaridas,
789', 'Brasília', 'DF', '34567-890', '1975-12-31');
```

Data Definition Language



```
SELECT * FROM clientes WHERE nome LIKE '%Silva%';  
INSERT INTO clientes (id, nome) VALUES (10, 'Maria Silva');  
UPDATE clientes SET email = 'maria.silva@email.com' WHERE id = 10;  
DELETE FROM clientes WHERE id = 10;
```


Obrigado!

@silvestre_lago
github.com/SilvestreLago

@x.s.s____
github.com/Xandersonsilva