

Prática 2N1 - Sintaxe do PHP

Disciplina: Programação para Web I
Professor: José Olinda

Aluno(a): _____

Vamos treinar a sintaxe do PHP. Você pode usar diversas estratégias para exibir as saídas de dados no navegador, e está livre para aplicar seus conhecimentos em HTML e CSS à vontade para tornar suas soluções mais apresentáveis e dinâmicas. O objetivo é que você não apenas resolva os problemas, mas também compreenda como PHP interage com a web.

Instruções para Entrega

- Crie um repositório no GitHub para armazenar seus códigos PHP.
- Para cada exercício, crie um arquivo PHP separado (ex: exercicio1.php, exercicio2.php, etc.) e organize-os em pastas, se desejar (ex: secao1/exercicio1.php).
- Envie seu projeto para o Github e, como resposta à atividade no Classroom, **informe a URL do repositório**.

Seção 1: Variáveis e Tipos de Dados

1. Registro de cliente:

- Crie uma variável para armazenar o nome completo de um cliente.
- Exiba uma mensagem de boas-vindas personalizada no navegador.

2. Controle de estoque:

- Defina uma variável para representar a quantidade em estoque de um produto.
- Exiba a quantidade atual de itens.

3. Medição de terreno:

- Armazene a área de um terreno com valores decimais.
- Mostre a área calculada.

4. Status de pedido:

- Crie uma variável booleana para o status de um pedido.

- Exiba o status em forma compreensível usando condicional.
5. Preço de venda e tipo de dado:
 - Defina o preço unitário de um item.
 - Use `var_dump()` ou `gettype()` para exibir o tipo da variável.
 6. Cálculo de orçamento:
 - Declare duas variáveis: `custoMaterial` e `custoMaoDeObra`.
 - Calcule e exiba o custo total.
 7. Endereço de entrega:
 - Crie variáveis para rua e número.
 - Exiba o endereço completo.
 8. Mudança de categoria:
 - Defina uma variável com código alfanumérico.
 - Atualize para um número e exiba antes/depois.
 9. Valor de π (Pi):
 - Declare uma constante ou variável com valor de Pi.
 - Caso decida usar constante, use o seguinte padrão:
 - `define("FOO", "algumacoisa");`
 - Exiba o valor com 4 casas decimais.
 10. Contagem de palavras de resumo:
 - Crie uma variável com um parágrafo. Para o IFCE, o resumo deve conter entre 350 e 500 palavras.
 - Conte os caracteres com função nativa.
 - Use a documentação oficial:
 - `https : //www.php.net/manual/pt_BR/language.types.string.php`

Seção 2: Estruturas Condicionais (*if, else, switch*)

11. **Controle de acesso:** Um sistema de portaria precisa verificar a idade de um visitante. Se for maior ou igual a 18 anos, exiba "Acesso Liberado"; caso contrário, "Acesso Negado".
12. **Classificação de temperatura:** Um sensor meteorológico registra a temperatura. Classifique e exiba se a temperatura é "Positiva", "Negativa" ou "Zero".
13. **Sistema de avaliação:** Com as duas notas de um aluno, calcule a média. Exiba "Aprovado" se a média for ≥ 7 , "Recuperação" se for entre 5 e 6.9, e "Reprovado" se for < 5 .
14. **Agendamento semanal:** Um aplicativo de agenda exibe compromissos. Dada uma variável numérica representando o dia da semana (1 para Domingo, 2 para Segunda, etc.), exiba o nome do dia correspondente.
15. **Escolha de bebida:** Um menu de cafeteria oferece opções de bebidas. Dada uma variável com o nome de uma bebida ("Café", "Chá", "Suco"), exiba uma mensagem como "Você escolheu [Bebida]". Se a bebida não estiver no menu, exiba "Bebida não disponível".
16. **Cadastro de usuário:** Ao preencher um cadastro, o usuário informa seu estado civil com um código (única letra). Exiba uma descrição para "C" (Casado/a), "S" (Solteiro/a), "D" (Divorciado/a) ou "O" (Outro).
17. **Alerta climático:** Com base na temperatura atual, emita alertas:
 - Abaixo de 0°C: "Alerta de Gelo!"
 - Entre 0°C e 15°C: "Clima Ameno"
 - Entre 15°C e 25°C: "Clima Agradável"
 - Acima de 25°C: "Atenção: Calor Extremo!"
18. **Preferência de cor:** Um site permite que o usuário escolha uma cor de tema. Se a cor escolhida for "Vermelho", "Azul" ou "Verde", exiba uma mensagem de confirmação. Caso contrário, sugira uma cor padrão.
19. **Cálculo de salário:** Uma folha de pagamento precisa determinar o salário base de um funcionário. Com base no cargo ("Desenvolvedor", "Designer", "Gerente"), defina e exiba o salário correspondente.
20. **Verificação de acesso (Login):** Um sistema de login precisa verificar se um usuário está ativo e se possui permissões de administrador. Apenas se ambas as condições forem verdadeiras, conceda o acesso total. Não implemente um sistema de login, apenas simule.

Seção 3: Estruturas de Repetição (for, while, do..while)

21. **Contagem regressiva simples:** Exibir uma contagem de 1 a 10 para simular a ordem de uma lista. Por estar no lado do servidor, essa contagem não será interativa [ainda], mostre apenas os números em ordem crescente. Use a criatividade para parecer uma contagem regressiva.
22. **Itens pares em lista:** Mostrar apenas os números pares de uma sequência de 2 a 20, como se fossem identificadores de produtos.
23. **Preparando para lançamento:** Exibir uma contagem regressiva de 5 a 1 antes de um "lançamento". Semelhante a anterior, mas juntando texto e número e decrescente.
24. **Processamento Limitado:** Processar uma lista de 100 itens, mas interromper o processamento assim que 50 itens forem processados para evitar sobrecarga.
25. **Exceção em relatório:** Exiba uma tabela com número de matrícula, nome completo e nota de PWEB1 de 10 alunos. Simule os valores, não use valores reais.
26. **Entrada de dados repetitiva:** Simular uma entrada de dados onde o programa continua "pedindo" informações até que um valor específico (0) seja "digitado" pelo usuário (use um array/vetor para simular as entradas). Implemente usando *do..while*
27. **Listagem de participantes:** Exibir todos os nomes de uma lista de participantes para um evento.
28. **Total de vendas:** Calcular o total de vendas de um mês, somando os valores diários do dia 1 ao dia 100. Use números aleatórios para cada valor, dentro de um intervalo de R\$ 50,00 a R\$ 5.000,00. Consulte a documentação: <https://www.php.net/manual/en/function.random.php>
29. **Ciclo de produção:** Simular um ciclo de produção que se repete enquanto a quantidade produzida for menor que 5. A cada ciclo, registrar e exibir a quantidade atual. Simular com array.
30. **Tabelas de referência:** Gerar e exibir as tabuadas de 1 a 5 para uma consulta rápida. Faça um HTML e CSS bonitinho.

Seção 4: Funções (*void* e com retorno, parâmetros e argumentos) e princípio DRY

31. **Mensagem de boas-vindas:** Criar uma função que exiba uma mensagem de boas-vindas padrão em várias páginas de um site, sem precisar de informações adicionais.
32. **Calculadora básica:** Desenvolver uma função que realize a soma de dois números quaisquer e exiba o resultado. Quer desafio? Exiba usando um layout destes: <https://www.cssacademy.com/blog/create-a-calculator-in-html-and-css> (não precisa do JS, isso é outra disciplina).

33. **Cálculo de custo total:** Implementar uma função que calcule e retorne o custo total de vários itens, recebendo o preço unitário e a quantidade. O resultado deve ser usado em um cálculo posterior.
34. **Média de notas:** Criar uma função que receba um conjunto de notas de um aluno e calcule a média final, retornando esse valor.
35. **Validação de Código:** Desenvolver uma função que verifique se um código numérico é par ou ímpar, retornando verdadeiro ou falso.
36. **Sistema de elegibilidade:** Refatorar a lógica de verificação de maioria (exercício 11) em uma função reutilizável, que receba a idade e retorne o status de elegibilidade. Já fez a 11 usando função? Pois refaça agora e faça melhor.
37. **Personalização de mensagem:** Criar uma função que gere uma saudação personalizada para um usuário, recebendo o nome dele como entrada.
38. **Formatação de preços:** Implementar uma função que receba um valor monetário e o retorne formatado corretamente para exibição (ex: R\$ 1.234,56).
39. **Cálculo de área de terreno:** Desenvolver uma função que calcule e retorne a área de um terreno retangular, dados seus lados.
40. **Log de eventos:** Criar uma função que registre uma mensagem de evento junto com o *timestamp* atual, útil para depuração ou histórico de ações. Veja a documentação https://www.php.net/manual/pt_BR/function.time.php

Seção 5: HTML + PHP, Funções Nativas (include, require, divisão de templates)

41. **Título dinâmico de página:** Em uma página web (HTML), defina o título da página usando uma variável PHP para que possa ser facilmente alterado.
42. **Mensagem de alerta padrão:** Crie um arquivo PHP separado contendo uma mensagem de alerta padrão. Em sua página principal, inclua esse arquivo para exibir o alerta sem copiar o código.
43. **Estrutura de site reutilizável:** Divida a estrutura básica de um site (cabeçalho, navegação, rodapé) em arquivos HTML/PHP separados. No arquivo principal, monte a página incluindo essas partes para manter a consistência.
44. **Cabeçalho personalizado:** No cabeçalho de uma página web, exiba um título que é definido dinamicamente no arquivo principal antes da inclusão do cabeçalho.
45. **Biblioteca de funções:** Mova suas funções utilitárias (como cálculo de média ou formatação de preço) para um arquivo PHP dedicado. Inclua esse arquivo (*helpers.php*) em todas as páginas que precisarem dessas funções.
46. **Tratamento de arquivos essenciais:** Demonstre a diferença entre `include` e `require` ao tentar carregar um arquivo de configuração vital que está ausente. Observe como o comportamento do programa muda.

47. **Arquivo de configurações:** Garanta que as configurações do banco de dados (host, usuário, senha) sejam carregadas apenas uma vez, mesmo que o código tente incluí-las múltiplas vezes.
48. **Prevenção de redefinição de funções:** Crie um arquivo com uma função global. Demonstre como `require` causa um erro se o arquivo for incluído mais de uma vez e como `require_once` evita esse problema.
49. **Layout de blog:** Monte uma página de blog usando múltiplos arquivos: um para o cabeçalho, um para o corpo do artigo, um para a barra lateral de categorias e um para o rodapé, todos combinados em um arquivo principal. É ruim de CSS? Pegue um modelo pronto aqui <https://html5up.net>
50. **Sistema de templates básico:** Crie um arquivo de template HTML com "espaços reservados" para o título e o conteúdo principal. Em um arquivo de lógica, defina o título e o conteúdo e, em seguida, use o template para renderizar a página final.