Recuperação de Férias

1. Começa aqui seu curso de PHP Moderno.

Resumo: Nesse primeiro vídeo, ele introduz como vai funcionar o curso e mostra como vai ser o primeiro módulo.

Explicando o primeiro módulo, ele mostra as etapas a serem seguidas durante essa etapa, e diz q vai tentar nos explicar da forma mais fácil possível, focando no raciocínio, para q possamos entender de uma maneira mais tranquila, mesmo n tendo um conhecimento prévio grande. E aconselha a buscarmos outros cursos, como CSS, HTML e JS.

1. Esse curso de PHP serve pra mim?

Resumo: Nesse vídeo ele mostra argumentações do pq vc deveria fazer o curso e coisas q podem fazer vc querer fazer o curso, como dúvidas e questionamentos em q o curso vai te ajudar. E mostra tbm para quem não é o curso.

No fim do vídeo ele faz um acordo rápido, pedindo para q possamos seguir esse acordo e estudar de vdd durante o curso e nos organizarmos para n nos perdemos.

1. Lista TOP 6 livros de PHP:

Resumo: Nesse vídeo ele fala sobre 6 livros q vão nos ajudar em nossos estudos. E explica q alguns livros n são recomendados para iniciantes e mostra eles, como o livro “PHP Moderno”.

1. A evolução do PHP:

Resumo: Nesse vídeo ele nos mostra a evolução do PHP como diz o título. Ele começa introduzindo um pouco sobre Rasmus Lerdorf, criador do Personal Home Page (PHP - 1994), ele tbm foi um dos colaboradores na criação do Apache e no MySQL.

Ele criou o PHP sem o intuito de fazer uma nova linguagem de programação, pois ele queria fazer apenas uma ferramenta simples q registrava quantos usuários acessavam seu site.

Mais tarde em 1995 ele oficializou isso e lançou a primeira versão do PHP (PHP Tools), ainda não era uma linguagem, que tinha o contador de usuário e adicionou um interpretador de usuário, onde acabou mudando de nome duas vezes.

Em 1996 ele lançou a versão 2.0, onde começou a ter mais cara de linguagem de programação. No mesmo ano após perceber q precisaria de ajuda para continuar, ele registrou o php como open source.

Em 1997 dois estudantes israelenses ao perceberem que o PHP n tinha tudo oq era necessário, se ofereceram para ajudar Ramus, assim, se tornando colaboradores.

Em 1998 depois de várias mudanças, foi lançado a versão 3.0, onde também mudaram novamente o nome do PHP (Personal Home Page) para PHP3 (PHP: Hypertext Prepocessor). Nesse mesmo ano duas pessoas importantes surgiram tbm, Colin Viebrock, que criou a logo do PHP, e Vincent Pontier, que criou a mascote do php (um elefante azul).

Em 1999 os estudantes israelenses criaram a zend, uma empresa especializada no desenvolvimento dessa linguagem.

Após tudo isso surgiram todas as outras versões do PHP, menos a 6.0. E descobrimos q o Rasmus n faz mais parte da equipe de desenvolvimento do PHP.

1. Por que um elefante é a mascote do PHP?

Resumo: Nesse vídeo ele explica do pq a mascote do PHP é um elefante. Ele começa falando q o PHP (Personal Home Page) iniciou como um nome e dps mudou para PHP (PHP: Hypertext Prepocessor). A mascote elePHPant surgiu pq o Vincent olhou para a sigla PHP e viu um elefante, sendo o “P” a cabeça e a tromba, o “H” o corpo, e o último “P” seria a traseira e o rabo do elefante. E mais pra frente ele diz tbm q escolheu o elefante por ser um animal muito inteligente e carismático.

1. As versões do PHP e seus recursos:

Resumo: Nesse vídeo ele resume oq veio em cada uma das versões, suas novas funções, e alguns problemas. Ele tbm explica q o PHP 6 nunca foi lançado oficialmente, apenas versões de teste.

O PHP 5.0 ficou 11 anos evoluindo e mudando e melhorando várias coisas, e em 2015 surgiu o PHP 7.0 que é 9x mais rápido que o PHP 5.0, além de trazer muitas funções novas q facilitaram muito na programação, e permitiu a utilização de outras bibliotecas no PHP.

Em 2020 surgiu o PHP 8.0, trazendo o Just-In-Time compilation, que detecta o trecho de um código, tipo o CPU Bound, executa ele de forma mais rápida jogando esse trecho de código para a memória da máquina, deixando o código bem mais rápido.

1. O PHP vai morrer? Vale a pena estudar PHP?

Resumo: Ele começa explicando q o PHP vai morrer sim, e as outras linguagens tbm vão, mas isso n deve fazer a gnt pensar q n compensa estudar sobre, pois cada um desses conhecimentos vai nos ajudar na programação.

E ele rebate algumas crenças ruins q as pessoas tem em relação ao PHP, mostrando dados, estatísticas e alguns exemplos.

1. Como funciona o PHP?

Resumo: Ele cmc falando q o PHP utiliza mto de HTML, CSS e JS, mas q funciona do lado do servidor, após isso ele fala sobre o cliente-side e o server-side, e da um exemplo de quando vc tenta acessar um site, e o servidor utiliza um número para ser acessado, e o servidor DNS (servidores de nomes de domínio faz isso) faz a ligação entre a URL e o número do servidor.

Ele fala também q o site feito em cliente-side é aquele site feito apenas com HTML, CSS e JS básico, onde o cliente tem mais trabalho, q após achar os dados buscados, ele envia para o cliente onde o computador vai processar e mostrar a informação buscada. E o server-side o servidor tem mais trabalho, ele n consegue enviar os arquivos para o cliente, pois a maquina n consegue executar o script, e precisa de um interpretador de PHP para q funcione.

Mais pro final ele fala da diferença entre os desenvolvedores front-end, back-end e full-stack. E após isso mostra outros exemplos de outras linguagens server-side, como python, Microsoft.Net, etc.

1. É possível criar qualquer coisa com PHP?

Resumo: Ele cmc falando q o PHP como outras coisas, serve para diversas coisas, mas não pra tudo. Ent o PHP vai te ajudar em diversas coisas, principalmente em uso de servidores, mas n em tudo, ele pode até fzr algumas coisinhas diferentes, mas n é a ferramenta pra isso, ent tem q saber com oq vai usar.

1. Como transformar seu computador em servidor PHP:

Resumo: Ele começa falando q vc precisa de um navegador e um editor de código(VS Code) para o cliente-side, e para o server-side, fala q vms precisar do apache, do mysql e de um interpretador PHP.

Ele fala também sobre o AMP, q é quando te o apache, mysql e o php junto, e para cada sistema operacional possui um pra cada, por ex no Linux q é o LAMP (L de Linux + AMP) e assim para todos os sistemas operacionais.

E nos apresenta após isso o XAMPP, q funciona para todos os sistemas operacionais.

1. É possível treinar PHP no smartphone:

Resumo: Nesse vídeo ele mostra um sandbox onde a pessoa pode treinar PHP, e indica vários q pode ajudar a pessoa a treinar, fala q eles n conseguem fazer tudo, mas ajudam bastante.

1. Seu servidor PHP no Windows:

Resumo: Nesse vídeo ele ensina vc a transformar sua maquina em um servidor, ele cmc ensinando a como baixar o XAMMP, explica q devemos ler cada etapa durante a instalação, e recomenda abrirmos como administrador, para q possamos usar com todas as funções funcionando. Após baixar tudo, ele fala q precisamos de um navegador para rodar, e recomenda o Chrome. Explica como deixar o XAMPP como administrador e indica baixar o VSCode para fazermos nossos códigos.

1. Seu servidor PHP no Linux:

Resumo: Nesse vídeo ele ensina a como transformar o Linux em um servidor PHP. Ele cmc mostrando como baixar o XAMPP, dps ele mostra como instalar o navegador Chrome, q é o que ele recomenda e por fim ensina a baixar o VSCode.

1. Seu servidor PHP no MacOS:

Resumo: Semelhante ao outros, nesse vídeo ele ensina a baixar a transformar o MacOS em um servidor PHP, cmc baixando o XAMPP, da as recomendações e os passo a passo de como baixar, dps ensina a baixar o Chrome e o VSCode.

1. Configurações importantes para VS Code:

Resumo: Ele cmc mostrando como achar a pasta htdocs e criar a sua primeira pasta que o servidor vai acessar, ensinando em cada um dos programas, Windows, no MacOS e no Linux. Ai ele mostra as configurações q ele julga importante, como o autosave, q ele configurou para salvar toda vez q troca de janela, ele também recomenda aumentar um pouco a fonte para facilitar a visão e mostra tbm o World Wrap, q faz as linhas de códigos extensas se quebrarem e irem para a linha de baixo, evitando aquela rolagem horizontal chata.

1. Melhores extensões para VS Code:

Resumo: Nesse vídeo ele mostra extensões q vão nos ajudar a trabalhar de forma mais rápida e q podem nos auxiliar durante a montagem dos códigos. Ele mostra 4 extensões muito boas, como a q traduz a interface, um q ajuda no preenchimento de códigos, outro q ajuda a formatar o HTML em PHP, evitando q algumas formatações percam suas funções e por fim um q te ajuda no preenchimento de códigos tbm, mas de forma completa em alguns casos.

1. Primeiro programa em PHP:

Resumo: Nesse vídeo ele ensina a gnt a fazer o primeiro código PHP, fala q o echo e o print funcionam, fala sobre algumas antigas formas de usar a tag PHP. E pede para a gnt fzr uma tarefa com ele. Ele também mostra que a nova versão do PHP disponibiliza a utilização de caracteres Unicode.

1. O PHP por dentro:

Resumo: Ele começa explicando oq é cada uma dos códigos base HTML, como o !DOCTYPE, q é o tipo de documento q esta sendo utilizado, nesse caso html, o head q é onde vc coloca as linkagens, define o tamanho da tela ocupada e o título(pode ser colocado mais coisas), e tem tbm a body q é o corpo do nosso site, onde podemos colocar as informações q irão aparecer para o cliente. Continuando ele fala sobre 5 formas de usar tags pra rodar o PHP, mesmo algumas n funcionando hj em dia, sendo elas: <?php?>, <??>, <%%>(forma q surgiu por conta do concorrente do php, mas se configurar certo roda php nele), e se tiver apenas uma linha <?php?> com o echo dentro, <?php echo”olá mundo”?> podemos substituir o php por =, mas somente nesse caso <?= echo”óla mundo”?>. E ele dá uns exercícios.

1. Como configurar o php.ini

Resumo: Ele fala sobre como resolver o problema que estava acontecendo no Mac e no Linux, pois o erro n mostrava, ent ele explica como mexer no php.ini, q define como o PHP vai se portar no servidor, ent mexendo nele, conseguimos fazer com q algumas funções sejam habilitadas e outras desativadas. E mexendo tbm nos permite ativar os shorts\_oppen\_tags <??>, mas ele explica q ficar usando elas a torto e a direita pode dar problema, pois n temos como saber se o servidor q o programa será enviado permite alterações, ent é melhor evitar de usar.

20. Variáveis e constantes em PHP:

Resumo: Ele começa explicando com um exemplo oq é uma variável e uma constante, a variável é como se fosse uma gavetinha dentro de um grande compartimento onde vc guarda a variável e pode alterá-la quando necessária, e a constante é quando n se pode alterar. E vemos q as “$”(variáveis) e as constantes(constante n tem símbolo) ficam alocadas no servidor, pois o PHP é uma linguagem server-side. Ele também passa 7 regras q são muito importantes para a as regras das variáveis e const. E passa umas recomendações de nomeação, como o camelCase (letra maiúscula toda vez q troca de palavra) e o Snake\_Case (usar \_ para separar as palavras).

21. Tipos primitivos do PHP:

Resumo: Ele fala sobre 3 tipos primitivos, mas da o foco principal aos escalares, e dá um exemplo:

$sobrenome = “Silva” – string (sequência de letras, números e símbolo dentro das “”)

$idade = 34 – int ou integer (números inteiros)

$peso = 85.9 - float, doable ou real (deixou de existir a partir do 7.4 do php); valores numéricos reais, com parte decimal.

$casado = true – bool ou boolean (valores lógicos, aceita apenas valores true e false)

E ele utiliza o var\_dump() e um momento do vídeo, que mostra todas as informações da variável escolhida.

Para uma variável boleano, escrever echo ou print, o valor vai ser 1 caso seja true, e caso seja false n vai aparecer nd.

E ele finaliza explicando os tipos primitivos compostos e especiais:

Compostos:

array: coleção de dados, de diversos tipos normalmente dentro de um [].

Object: serve para guardar uma coleção de propriedades cada uma com sua chave única.

22. Manipulação de strings com PHP:

Resumo: Ele começa explicando sobre a concatenação de strings “.”, no caso do JS e outras linguagens é utilizado o “+”, só q no PHP é utilizado o “.” e o “+” só serve para somar números.

As aspas duplas “” interpretam o conteúdo e geram um resultado.

As aspas simples ‘’ não interpretam o conteúdo, então ele gera o resultado igual foi escrito no código.

Por mais q uma cont esteja dentro de uma aspa dupla, ela n vai ser interpretada por n ter um $, ent ela n faz a interpolação (troca do nome do elemento pelo conteúdo).

E ele fala q nem sempre a ausência de uma mensagem de erro, n significa q o código está correto.

Sequência de escape, usada para colocar “” dentro de uma string

Ex:



As aspas duplas possuem várias sequências de escape, como: \n (nova linha); \t (tab); \\ (barra invertida); \$ (cifrão) e \u{} (usado para interpretar códigos Unicode), mas a simples só possui o \’conteudo\’.

Heredoc = gerada em varias linhas, quebra de linhas só em formato texto. Ela também interpreta o código todo, e para abrir esse tipo de formatação, é preciso fzr isso

Echo <<< ALGO (pode ser qualquer coisa, mas tem q ser em maiúsculo)

Conteúdo aqui

Conteúdo aqui

ALGO;

Nowdoc = quase a mesma coisa, mas diferente da heredoc, ela n possui interpretação de código, e coloca-se ‘’ dentro do nome de abertura, ex: ‘ALGO’.

23. Obtendo dados de formulários com PHP (parte 1):

Resumo: ele explica como fazer um formulário com PHP. Ele começa falando sobre coisas q n podemos esquecer, como o method (como serão codificados para serem enviados ao servidor) e action (para onde vão ser enviados); ele também fala um pouco sobre os métodos de envio, o $\_GET e o $\_POST, no $\_GET a url aparece, e no $\_POST não. Há mais coisas sobre os métodos, mas ele n foca nisso nesse video

24. Obtendo dados de formulários com PHP (parte 2)

Resumo: nessa segunda parte ele quer enviar os dados do formulário para outra página, e nos apresenta o method $\_REQUEST, q é uma junção de $\_GET, $\_POST e $\_COOKIES. Ele também nos mostra um novo operador, o operador de coalescência nula “??”, que caso n tenha nada nos campos preenchidos ele coloca uma mensagem no lugar onde ficaria no nome ($nome = $\_GET[“nome”] ?? “?”;) o ? dentro dos “” vai ser exibido no lugar do nome que seria enviado.

Nesses dois vídeos ele ensina a como pegar dados de formulários e enviar ele para outra área, usando superglobais.

25. Expressões Aritméticas com PHP:

Resumo: Nesse vídeo ele da um foco nas expressões compostas. Ele explica sobre a ordem de precedência, onde cada expressão de cálculo possui uma ordem, e sempre a com maior ordem é a primeira a ser calculada. São elas (em ordem de precedência já): \*\*(elevado, apenas a partir da versão 5.6 do PHP), \*(multiplica), / (divisão real), % (mostra o resto de uma divisão inteira, módulo), + (adição) e – (subtração).

Ex: 50/2+3\*\*2 = 34, pois 3\*\*2 = 9, dps 50/2 = 25 e por fim 25+9 = 34

Mas utilizando os (), {} e [] podemos mudar as ordens de precedência, se houver () calcula-se primeiro o que está dentro deles, depois oq está dentro das chaves {}, ai por fim oq está dentro dos colchetes [], e caso tenha números fora de tudo isso, ai por último se faz eles.

26. Funções Aritméticas do PHP:

Resumo: Ele da continuidade a aula anterior, mas explicando como suprir as funcionalidades de operadores q n existem. Ele começa falando sobre a função abs(), retorna valor absoluto de um número; tem tbm o base\_convert(), que converte números de bases octa, deca decimal e binário, além de outros;

ceil(), floor() e round(), q servem para arredondar números decimais, ceil pra cima, floor pra baixo e round de forma aritmética;

hypot(), calcula a hipotenusa;

intdiv(), divisão inteira, ele basicamente pega o número mais próximo da divisão, por ex: 5/2 = 2,5 nesse caso ele vai pegar apenas o 2, pois o 1 q sobra para continuar a divisão n será usado;

min() e max(), servem para definir o maior ou menor valor de uma sequência de números;

pi(), função do valor pi, mas também podemos usar a constante M\_PI ;

pow(), potenciação (era utilizado antigamente, pois n existia o \*\* como função para isso, ainda existe, mas quando usado ele perde a ordem de precedência);

sin(), cos() e tan(), calcular seno, cosseno e tangente;

sqrt(), calcula a raiz quadrada de um número;

27. Operadores Aritméticos do PHP:

Resumo: nesse vídeo ele começa com uma dúvida sobre se “2” + “2” = ?, ent ele explica, que no PHP o + é o operador aritmético de adição, ent ele não vai fazer igual o JS que nesse caso juntaria as strings ficando “22”, mas no PHP ele vai fzr o cálculo e vai fica 4 como número inteiro, pois no PHP o + vai buscar os números q precisam ser somados dentro da string, e caso queira “22” vc deve usar a concatenação . para juntar eles, ent “2” + “2” = 4. Entretanto nas versões antigas quando vc usava + para somar dois nomes dentro de strings, ele dava o valor 0, pois n achava nenhum número e colocava 0, mas se tivesse um número dentro dessa string, ele ignorava as palavras e somava apenas os números, mas para q a soma funcione, ambos números precisam estar na mesma posição, senão ele pega apenas o que está na primeira posição.

28. Cinco desafios PHP para iniciantes:

Resumo: ele passa 5 desafios para praticarmos e trabalhar as coisas q ele nos ensinou durante o curso, focando na questão da aritmética, e explica q vai nos apresentar outros comandos durante as resoluções dos exercícios.

29. Desafio 001: Antecessor e Sucessor:

Resumo: montamos uma área onde vc escolhe um número e na outra mostra o sucessor e o antecessor do número escolhido.

30. Desafio 002: Sorteador de números:

Resumo: antes de resolver o exercício ele fala sobre dois tipos de funções para gerar números aleatórios, a rand() e a mt\_rand(), na qual a mt\_rand é mais rápida e moderna em comparação com a rand(). Entretanto, o mt\_rand() precisa que o primeiro valor colocado seja maior q o segundo valor. Ele também mostra o random\_int(), que gera números aleatórios criptografados de forma segura, mas é extremamente lento.

E não necessariamente vc precisa criar uma variável para armazenar o número do rand, mas pode colocar os números q deseja dentro dele já, tipo assim: mt\_rand(0, 100);

31. Desafio 003: Conversor de Moedas básico:

Resumo: Nesse desafio ele ensina a fazermos o conversor de real em doar utilizando uma cotação do dólar fixa.

No começo ele mostra um comando para definirmos as casas decimais e podermos decidir qual é o símbolo de separação de casas de milhar e decimal, q é o number\_format($variavel, 2(casas decimais), “,”(separação da casa decimal), “.”(separação da casa de milhar));

Ele também passa um código muito importante para esse tipo de coisa, q é meio q uma biblioteca, q é esse $padrao = numfmt\_create("pt\_BR", NumberFormatter::CURRENCY); que faz com que os números colocados sejam formatados de acordo com a localidade colocada dentro do “”, tipo “pt\_BR” q é como o brasil vê os valores em reais e outras moedas. E precisa de um acompanhante, q é o numfmt\_format\_currency($padrao, $money, "BRL"); e o “BRL” é como será mostrado a formatação das moedas.

32. Desafio 004: Conversor de Moedas avançado:

Resumo: é basicamente como o outro, mas nesse ele ensina a gnt a pegar uma API para podermos utilizar, assim a cotação sempre vai estar vigente com a atual.

33. Desafio 005: Analisando um número Real:

Resumo: No começo n tinha entendido como pegaria a parte fracionada, mas dps percebi q era só utilizar o number\_forat() para definir as casas decimais q iriam aparecer e fazer um calculo lógico do número escolhido – a parte inteira dele, que daria a parte fracionada. E dá para vc definir q quer somente o número inteiro colocando um (int) atrás da variável, forçando-a a mudar sua primitividade.

34. Superglobais no PHP:

Resumo: Nesse vídeo ele vai nos ensina sobre as superglobais e suas funções. Durante o teste da $\_GET, ele fala sobre a query string, que é um conjunto de valores anexos a URL. Ele também mostra que o $\_POST n é tao seguro quanto pensam, pois da para ver os dados se vc procurar, a diferença dele pro GET é q ele da um pouco mais de trabalho para achar.

$\_GET: o GET pega dados contidos na URL do programa;

$\_POST: o POST pega apenas pelo cabeçalho de um formulário;

$\_REQUEST: e o REQUEST pega de ambos, seja da URL ou dos formulários;

$\_COOKIE: são pequenas variáveis guardadas dentro de seu computador, que o usuário pode apagar a qualquer momento, e elas tem acesso apenas aos cookies da sua aplicação PHP, e elas servem para guardar algumas preferências ou saber se o usuário está logado;

$\_FILES: pega arquivos que foram feitos upload;

$\_SESSION: guarda dados temporários de cada usuário até a q sessão expirar, e serve para manter os dados entre as navegações de páginas;

$\_ENV: é uma variável de ambiente, mostra variáveis do ambiente.

$\_SERVER: mostra as variáveis do servirdor;

$GLOBALS: ela mostra as informações de todas as superglobais;

Mais pro final ele explica do pq utilizou a tag <pre></pre>, que serve para organizar os arrays, e deixar mais fácil de ler e identificar, pois separa eles em linhas, e é muito utilizado quando usamos o var\_dump(). E toda vez q vc rodar um script, elas vão estar presentes, mesmo que sem nenhuma informação.

35. Formulários retroalimentados:

Resumo: ele começa explicando oq é esse formulário retroalimentado, que basicamente, ao invés de sair de um formulário HTML e ir para um PHP, ele vai fazer todo esse processo apenas em uma única página PHP. Ele relembra do uso da PHP short tags (utiliza <?= no lugar de <?php, quando tem apenas um único echo no código).

Para q o action do form possa ir para a própria página, ele recomenda utilizar o <?= $\_SERVER['PHP\_SELF'] ?>, dentro do action, que pode ser achado dentro da superglobal SERVER.

36. Oito desafios PHP para iniciantes:

Resumo: Nesse vídeo ele passar mais 8 desafios para podermos praticar eles, em sua maioria envolve aritmética e estão em um nível um pouco assim.

37. Desafio 006: A anatomia de uma divisão:

Resumo: Nesse vídeo ele meio q mostra como fazer um Analisador de divisão inteira, que mostra o dividendo, divisor, produto e sobra. E usa basicamente os mesmos conceitos do exercício 006 da aula 35.

38. Desafio 007: Salário Mínimo com PHP:

Resumo: Ele mostra como fazer a comparação de determinado salário com um salário mínimo, por ex: tal salário equivale a tantos salários mínimos.

Ele começa explicando sobre a divisão inteira e real. A real é aquela divisão que enquanto sobrar vai acontecer a divisão, e pode acontecer o caso da dizima periódica, q é quando a divisão n tem fim. E a inteira vc para de dividir a partir do momento em q não tem mais como dividir de forma inteira, assim obtendo o resto da divisão.

Ex: Real: 8/3 = 2,666... (com dizima) ou 5/2 = 2,5 (sem dizima)

Inteira: 8/3 = 2 com resto 2 ou 5/2 = 2 com resto 1

39. Desafio 008: Exercício das Raízes:

Resumo: Ele mostra como fazer um calculador de raiz quadrada e cubica, utilizamos o sqrt() para podermos calcular APENAS raiz quadrada, e para a cubica calculamos utilizando a potenciação de uma fração, desse jeito $numero \*\*(potenciação) (1/3) raiz cubica e assim para outras raízes. Ele no começo do vídeo explica como funciona a raiz quadrada e cubica.

40. Desafio 009: Entendendo as Médias:

Resumo: Nesse vídeo ele ensina a fazer um calculador de média ponderada e aritmética simples, ele explica elas no começo, sendo a aritmética a soma dos valores dividido pela quantidade (20+20+20/3= 20), a soma dos itens vem primeiro e depois a divisão. E a ponderada é com a multiplicação do valor com o peso dele dividido pela soma dos pesos (vl1\*p1 + vl2\*p2/p1+p2=x) primeiro faz a multiplicação, depois a soma e finaliza coma divisão.

Ele ensina a como selecionar certos itens específicos e escrever uma mesma coisa em lugares diferentes, segurando a tecla Alt e depois escrevendo oq quer escrever.

41. Desafio 010: Máquina do Tempo:

Resumo: Ele ensina a como fazer um calculador q vai mostrar sua idade em determinada época de acordo com o ano em q vc nasceu.

42. Desafio 011: Reajustar Preços:

Resumo: Ele começa falando sobre como funciona a divisão dando um ex de um quadrado de 100m², e que dependendo da forma como vc divide ele, vc tem uma porcentagem diferente, por ex, de vc divide na metade, fica dois retângulos de 5x10m q da 50m², ou seja vc dividiu em 2 partes ent vc tem 50%, pq são 2 partes e 50/100 é 2, e assim para qualquer outra porcentagem. Ele depois explica como fazer a porcentagem utilizando a regra de 3.

Ex: 500 --- 100% 500X = 250 \* 100 X = 25000/500

250 --- X% = 50% 500X = 25000 X = 50%

Ele mostra o input do tipo range, q é uma barrinha com um círculo móvel dentro dela, e é com ele q fazemos os reajustes das porcentagens, e para a porcentagem acompanhar na tela o mesmo do range sem ter q ficar reiniciando a tele, ele aplica uma função JS para ajudar no visual. Onde ele iguala o id do símbolo q fica no lugar do número, antes dele ser escolhido, com o valor da porcentagem, assim atualizando na tela mesmo toda vez q vc move o range.

43. Desafio 012: Calculadora de Tempo:

Resumo: Ele começa mostrando como funciona o princípio da divisão do tempo, de segundo pra minuto, de segundo pra hora e assim por diante até chegar em semana. E depois ele mostra aplicando isso com um ex, 2.000.00seg equivale a 3 semanas, 2 dias 3 horas, 33 min e 20seg. pois vc pega os 2M seg e divide pelo valor da semana em segundos, com a sobra vc divide com tempo em segundo do dia e assim por diante.

44. Desafio 013: Simulando um Caixa Eletrônico:

Resumo: Nesse vídeo ele ensina a como fazer um caixa eletrônico, onde vc coloca o valor q quer sacar e em baixo vai mostrar quais notas serão entregues a vc, priorizando a menor quantidade de cédulas. E utilizamos comandos semelhantes ou iguais ao desafio anterior para mostrar a quantidade de cada cédula. E utilizamos o step para definir de quantos em quantos o valor pode ser escolhido, no nosso caso, um step de 5, sendo assim, apenas valores que terminem com 5 ou 0 serão aceitos.

2. Pesquisa sobre os seguintes conteúdos:

Você deve pesquisar e explicar com suas palavras o que aprendeu sobre cada um dos itens abaixo. Dê exemplos de código para cada um.

**Superglobais:**

$\_SESSION

Funciona de forma temporária, apenas enquanto o usuário está na página web, e serve para salvar as informações do usuário enquanto ele tá logado, e tende a ser mais seguro.

<?php

session\_start();

$\_SESSION['usuario'] = 'Luiz';

echo "Olá, " . $\_SESSION['usuario']

$\_POST

Difente do $\_GET, que pega os dados da URL, p $\_POST pega os dados enviados por um formulário, sendo um pouco mais seguro, mas ainda sim tem como achar as informações mandadas pelo método $\_POST.

<form action="index.php" method="post">

<label for="text">digite seu nome aqui</label>

<input type="text" name="text" id="text">

</form>

<?php

$nome = $\_POST['text'];

echo "Olá $nome"

?>

$\_COOKIE

Semelhante ao $\_SESSION, ele pode ser usado para verificar se o usuário está ativo, entretanto essa verificação pode durar de acordo com o tempo colocado no código. Ele não é tão seguro igual o $\_SESSION por ser público, mas é muito bom para salvar dados de forma maleável, pq vc pode controlar o tempo q o $\_COOKIE vai durar.

<?php

//cria o cookie

setcookie("usuario", "Luiz", time() + 1800); //tempo defifinido em segundos

//executa ele

isset($\_COOKIE['usuario']);

echo "bem vindo, " . $\_COOKIE['usuario']

?>

**Armazenamento no navegador:**

LocalStorage (JavaScript)

**Funções importantes:**

isset(),

verifica se a variável existe, ou se tem algum valor diferente de nulo.

<?php

$nome = "luiz";

if (isset($nome)) {

    echo "<p>A variável 'nome' existe!";

} else {

    echo "<p>A variável 'nome' não existe!";

}

empty()

verifica se a variável não possui nada de relevante, e verifica se a variável está vazia.

$idade = 0;

if (empty($idade)) {

    echo "<p>A variável 'idade' está vazia!";

} else {

    echo "A variável 'idade' tem valor!";

}

?>

filter\_input(),

pelo q entendi ele vai filtrar o conteúdo do form de acordo com o method escolhido, ent pega o formulário, pega o campo do formulário que quer e depois valida ele, caso seja valido retorna true, senão retorna false.

 <form method="post"> <!-- metodo escolhido -->

  <label>Email:</label>

  <input type="text" name="email"> <!--name é o formulario q vc vai querer pegar-->

  <input type="submit" value="Enviar">

</form>

<?php

$email = filter\_input(INPUT\_POST, 'email', FILTER\_VALIDATE\_EMAIL); //pegou o method, depois o nome do formlario e por vim fez a validação

if ($email) {

  echo "O email $email é valido";

} else {

  echo "Email inválido!";

}

?>

filter\_var()

ele tem varias funções, muito uteis inclusive, principalmente pq dá para validarmos números, datas e até mesmo ver se um email está formatado corretamente. É basicamente um fiscal de informações.

<?php

$email = "luiz@gmail.com";

if (filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

  echo "Email válido!";

} else {

  echo "Email inválido!";

}

htmlspecialchars()

serve como forma de proteção contra códigos maliciosos, pois ele pega o código e transforma em uma string html, e o código aparece a tela da mesma forma q é escrito no programa.

<?php

$str = "isso é um texto em <b>bold</b>.";

echo htmlspecialchars($str);

array(),

transforma conjuntos de strings em únicos, e da uma numeração a eles, de 0 em diante.

<?php

$animais = array("cavalo", "galinha", "boi");

echo $animais [0];

echo $animais [1];

echo $animais [2];

foreach

é como se fosse um dono do array, que ele pega as informações do array e vai mostrando uma por uma.

<?php

$idades = array("10", "29", "12");

foreach($idades as $idade){

    echo $idade . "<br>";

}

array\_push(),

ele adiciona novos itens a um array, aumentado o tamanho dele, por ex: se tem 2 itens num array, vc seleciona a variável q estão esses 2 itens e coloca mais dois utilizando o array\_push().

<?php

$fazenda = ["cavalo", "boi"];

array\_push($fazenda, "vaca", "porco");

print\_r($fazenda)

in\_array()

ele serve como um verificador de elementos dentro de uma array, pois ele verifica se o item selecionado existe ou não dentro de determinada array.

<?php

$fazenda = ["boi", "porco", "abelha"];

if (in\_array("vaca", $fazenda)) {

    echo "Vaca está na lista!";

} else {

    echo "Vaca não está na lista!";

explode(),

pega strings que estão juntas e separa elas, transformando em um array.

$animais = "vaca,porco,galinha,peixe";

$celeiro = explode(",", $animais);

echo "<pre>";

var\_dump($animais);

var\_dump($celeiro);

echo "</pre>";

implode()

é o contrário do explode, aqui o include pega arrays e transforma em uma única string.

$animais = ["vaca","porco","galinha","peixe"];

$celeiro = implode(",", $animais);

echo "<pre>";

var\_dump($animais);

var\_dump($celeiro);

echo "</pre>";

strlen(),

ele é bem simples, ele basicamente é usado para contar a quantidade de caracteres de um texto, ou contidos dentro de uma variável.

<?php

$texto = "Olá, mundo!";

$tamanho = strlen($texto);

echo "O texto tem $tamanho caracteres.";

echo "<p>O texto era '". $texto ."'";

str\_replace(),

utilizado para inverter ou trocar palavras sem que vc precise alterar o texto todo, é muito útil para trocar palavras inapropriadas entre outras coisasa.

<?php

$frase = "Eu gosto de chocolate.";

$novaFrase = str\_replace("chocolate", "baunilha", $frase);

echo $novaFrase;

substr(),

ele basicamente pega patês de um texto, dependendo da posição, do texto q vc quer e da quantidade de caracterers q vcquer, é bom para quando vc quer abreviar o nome de alguém.

<?php

$nome = "Luiz Fernando";

$parte = substr($nome, 5, 3); //5 pois é onde esta o espaço de um nome para o outro, ent 3 crct serão escolhidos dali pra frnt

echo $parte;

strpos()

ele é semelhante ao anterior, ele pega uma palavra, letras ou números dentro de uma frase ou palavra e define sua posição a partir da posição da primeira letra.

<?php

$frase = "Eu gosto de abacate com amendoim";

$posicao = strpos($frase, "amen");

echo $posicao;

**Importação de arquivos:**

Include

Ele basicamente inclui um arquivo externo dentro de as pagina, ele traz as informações da outra pagina para ele mesmo, como se ambas fossem juntas.

include("ola.php");

echo "olá mundo!"

Require

Bem semelhante ao include, porem caso esse arquivo q foi inserido n existir, ele da um erro fatal e para todo código, já o incude não ele apenas avisa o erro e mantem o resto do codigo.

require("ola.php");

echo "olá mundo!"

include\_once

é bem semelhante a seu irmão, porem a diferença é q esse, o arquivo buscado só pode ser chamado uma única vez, mesmo q tentemos dnv. Caso n ache o arquivo ou de erro, ele emite uma mensagem de erro, mas o resto do código continua.

<?php

include\_once 'teste.php';

include\_once 'teste.php'; // este será ignorado pq ja foi chamado antes

require\_once

mesmo do seu irmão, mas pode ser buscado apenas uma vez, e como seu irmão, caso o código esteja errado, ou n exista o arquivo buscado, ele dará um erro fatal e vai parar todo programa.

<?php

require\_once 'teste.php';

require\_once 'teste.php'; // este será ignorado, pois ja foi chamado antes

1. O que são variáveis superglobais? Cite outras além das já listadas.

R: São variáveis especiais, que estão disponíveis em um script, elas n precisam de permissão ou q definimos sua função, elas mesmas já servem para isso, e servem em sua maioria para guardar ou coletar informações.

1. Qual a diferença entre isset() e empty()?

R: Ambos são funções que servem para verificara variáveis, a diferença entre elas se consiste basicamente em q o isset() vai ver se a variável existe ou se ela tem algum valor diferente de null(nulo), e o empty() vai verificar se a variável está vazia, sem nenhum valor completo dentro, ou se ela não tem nenhum valor de utilidade

1. Para que serve htmlspecialchars()?

R: Ele serve como um escudo, um protetor q impede que códigos maliciosos sejam processados, eles são executados apenas da maneira como são escritos. Ele basicamente converte caracteres especiais em strings html.

1. Explique a diferença entre explode() e implode() com exemplo.

R: Em resumo, pelo q entendi, o explode() pega elementos que estão juntos e separa eles, transformando ele em um array(lugar que armazena várias informações). E o implode faz o contrário, ele pega o array e transforma em uma string, tudo junto.

$animais = “vaca,porco,galinha,peixe”;

$celeiro = explode(“,”, $animais);

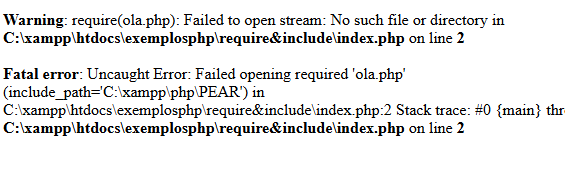
Ai para mostrar o resultado podemos usar o var\_dump($celeiro), que vai mostrar o array que o explode fez.

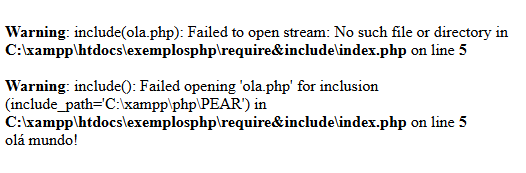
$animais = [“vaca”, “porco”, “galinha”, “peixe”]

$celeiro = implode(“,”, $animais);

Da mesma forma usamos o var\_dump($celeiro) para mostrar, só q dessa vez como string, e não array.

1. O que acontece se usar require para incluir um arquivo que não existe? E se usar include?

R: Em suma, o require vai parar todos o código, devido a um erro fatal, pois como ele n conseguiu achar o que estava procurando, ele para e todo o programa para junto dele. 

Já o include, aparece um aviso que algo esta errado, mas o resto do programa continua funcionando. Podemos ver no exemplo abaixo, que mesmo com o aviso, a mensagem de olá mundo aparece, diferente do exclude.

1. O que é LocalStorage e quando ele é mais útil que cookies ou sessões?

R: Ele é um espaço de armazenamento dentro do pc, serve para armazenar informação, como se fosse um bloco de notas. E ele é mais utilizado no lugar desses dois quando se trata de dados simples e não precisa enviar dado para o sevidor.

1. Em um sistema de login, quando você usaria $\_SESSION e quando usaria $\_COOKIE?

R: Eu usaria o $\_SESSION para verificar se tem algum usuário conectado, e para poder salvar as informações do usuário enquanto ele navega entre as páginas.

E usaria o $\_COOKIE para salvar temporariamente dados acessados pelo usuário, para poder fornecer para ele anúncios de coisas semelhantes as q ele procura.