HTML

Forms

Data sending ways:

1) GET

This is the default method of sending data in forms. Datas will send in URL

2) POST

Data is not send by URL, but you can see the data inside browser inspect, network, clicking in the archive (php f.ex.), and headers.

If the data is not sensible, you can use GET, but if you send data like password, you’ll ned POST.

GET can send maximum 3000 bytes, cannot send archives or photos.

Input configs

Required : This input is required to advance to next step

Minlenght : Minimum length size for data

Maxlength : Maximum length for data size

Placeholder : Content inside input

Autocomplete : a lot of options, like new-password, username, address, etc.

*TIPOS DE INPUT PARA FORMULÁRIOS:*

checkbox:

O input do tipo checkbox define uma caixa de seleção. A caixa de seleção é mostrada como uma pequena box quadrada que pode ser marcada pelo usuário com um clique (marca de verificação). As caixas de seleção são usadas para permitir que o usuário selecione *uma ou mais opções* dentre um número limitado de opções. A possibilidade de selecionar *mais de um item* AO MESMO TEMPO é uma característica fundamental do input checkbox. Sendo que somente os valores das caixas marcadas pelo usuário que serão enviados para o back-end quando a submissão do form for executada. Dependendo do seu projeto, você pode trabalhar com uma ou mais caixas de checagem no seu formulário. A quantidade de caixas de checagem (opções) contidas no seu form vai depender da necessidade da sua aplicação. O checkbox é um input de entrada como qualquer outro, portanto ele também precisa de uma label. Geralmente, para esse tipo de controle, no HTML nós inserimos o código do input ANTES do código da label, dessa forma a caixa de seleção fica na frente da label... No entanto, você também pode fazer o inverso e inserir o código do input após o código da label, posicionando a label na frente da caixa de seleção. Atenção: No input checkbox, os atributos "name" e "id" também devem ser configurados normalmente... No entanto, quando você tem mais de um checkbox no seu form e essas caixas de seleção representam diferentes opções dentro de um mesmo contexto, é recomentado atribuir a mesma nomenclatura para os atributos "name" dessas caixas de seleção, assim especificamos que essas boxes são de um mesmo grupo e portanto elas representam diferentes opções dentro de um mesmo assunto. Assim quando o usuário selecionar as opções de seu interesse, esses dados serão enviados para o servidor identificados com o mesmo nome, indicando que fazem parte de um mesmo grupo checkbox. Dessa forma, os atributos "name" são sempre iguais para cada grupo de checkbox... Por exemplo: Se no seu form você tiver DOIS GRUPOS de checkbox, para dois tipos de informações que não são correlacionadas, aí sim um desses grupos vai ter um name e o outro grupo um name diferente, desvinculando os grupos (já o id é sempre único para cada checkbox individualmente). Sendo que o VÍNCULO entre o componente checkbox e sua respectiva label também deve ser estabelecido. Uma outra característica *importantíssima* do input checkbox: Como sabemos, diferentemente de outros inputs, no checkbox o usuário não insere um dado digitando, ele simplesmente marca a opção que lhe atende... Dessa maneira, para o funcionamento correto de uma caixa de seleção, é necessário que especifiquemos um atributo "value" para a mesma... (insira o value dentro da tag do input) Determinando assim o valor (DADO) que aquela caixa de seleção REPRESENTA. O conteúdo contido nesse value que será enviado para o servidor no par nome-valor na hora da submissão do form (num checkbox: par nome-valor = atributos 'name' e 'value' respectivamente). Obs.: Se você não definir um atributo value para o checkbox, o dado que será enviado contido no par nome-valor será um dado genérico "on". Segue um exemplo de código com inputs do tipo checkbox:

<form action="cadastro.php" method="get"> <fieldset> <legend>Qual dessas é a sua linguagem de programação preferida?</legend> <input type="checkbox" name="linguagem" id="js" value="JavaScript"> <label for="js">JavaScript</label> <input type="checkbox" name="linguagem" id="php" value="PHP"> <label for="php">PHP</label> <input type="checkbox" name="linguagem" id="python" value="Python"> <label for="python">Python</label> </fieldset> </form>

Dessa forma, o checkbox ou caixa de seleção nada mais é que um tipo de campo que já contém informações predefinidas e que o usuário deverá marcar, dispondo da possibilidade de ativar e desativar valores individuais... Tendo a opção de selecionar nenhuma, uma ou mais caixas (somente os valores das caixas marcadas que serão enviados para o back-end).

radio:

O input do tipo radio define um radio botton (botão de rádio). O botão de rádio é mostrado como um pequeno círculo que pode ser selecionado pelo usuário com um clique. Esse elemento geralmente é usado em grupos (radio groups)... Ou seja, vários botões de rádio juntos formam um radio group (grupo). Os radio buttons são usados para permitir que o usuário selecione *uma opção* dentre um número limitado de opções. A possibilidade de selecionar APENAS UM ITEM dentre os disponíveis em um grupo é uma característica fundamental do input radio. Ao clicar em uma opção quando outra já está marcada, faz com que aquela opção marcada se desmarque automaticamente. Você pode ter vários grupos distintos de radio buttons no mesmo formulário, podendo fazer a seleção de apenas um item em cada um desses grupos. Nota: Uma característica peculiar do input radio é que quando uma opção é selecionada, se clicarmos novamente, a seleção não sai... Dentro de um grupo, você até pode clicar em outra opção para desmarcar a atual, mas a partir do momento que uma opção é selecionada, não é mais possível deixar as opções do grupo em branco novamente sem ter que recarregar a página (fora isso a função do input "reset" também consegue deixar as opções em branco novamente). Isso aparentemente ocorre porque os radio buttons teoricamente foram concebidos para não serem deixados em branco. Sendo que somente o valor do botão selecionado pelo usuário que será enviado para o back-end quando a submissão do form for executada. Dependendo do seu projeto, você pode trabalhar com um ou mais botões de rádio no seu formulário. A quantidade de radio buttons (opções) contidas no seu form vai depender da necessidade da sua aplicação. O radio é um input de entrada como qualquer outro, portanto ele também precisa de uma label. Geralmente, para esse tipo de controle, no HTML nós inserimos o código do input ANTES do código da label, dessa forma a caixa de seleção fica na frente da label... No entanto, você também pode fazer o inverso e inserir o código do input após o código da label, posicionando a label na frente da caixa de seleção. Atenção: No input radio, os atributos "name" e "id" também devem ser configurados normalmente... No entanto, quando você tem mais de um radio button no seu form e esses botões representam diferentes opções dentro de um mesmo contexto, é obrigatório atribuir a mesma nomenclatura para os atributos "name" desses radio buttons, assim especificamos que esses botões são de um mesmo grupo e portanto eles representam diferentes opções dentro de um mesmo assunto. No caso dos radio buttons, é muito importante que você faça como foi orientado acima e atribua a mesma nomenclatura para os atributos "name", definindo um grupo. A característica fundamental desse tipo de input é que dentre as opções disponíveis, o usuário só possa escolher uma... Entretanto, se você não definir uma mesma nomenclatura para os atributos "name" dos inputs que supostamente formam aquele grupo, a funcionalidade de escolher apenas uma opção é comprometida e anulada (passando a poder escolher 'mais de uma' opção), visto que o sistema considera esses inputs com names diferentes como grupos diferentes. Dessa forma, os atributos "name" são sempre iguais para cada grupo de radio buttons... Por exemplo: Se no seu form você tiver DOIS GRUPOS de radio buttons, para dois tipos de informações que não são correlacionadas, aí sim um desses grupos vai ter um name e o outro grupo um name diferente, desvinculando os grupos (já o id é sempre único para cada radio button individualmente). Sendo que o VÍNCULO entre o componente radio e sua respectiva label também deve ser estabelecido. Uma outra característica *importantíssima* do input radio: Como sabemos, diferentemente de outros inputs, no radio o usuário não insere um dado digitando, ele simplesmente seleciona a opção que lhe atende... Dessa maneira, para o funcionamento correto de um radio button, é necessário que especifiquemos um atributo "value" para a mesmo... (insira o value dentro da tag do input) Determinando assim o valor (DADO) que aquele radio button REPRESENTA. O conteúdo contido nesse value que será enviado para o servidor no par nome-valor na hora da submissão do form (num radio: par nome-valor = atributos 'name' e 'value' respectivamente). Obs.: Se você não definir um atributo value para o radio, o dado que será enviado contido no par nome-valor será um dado genérico "on". Segue um exemplo de código com inputs do tipo radio:

<form action="cadastro.php" method="get"> <fieldset> <legend>Escolha a opção que lhe atende:</legend> <input type="radio" name="opcao" id="masculino" value="M"> <label for="masculino">Masculino</label> <input type="radio" name="opcao" id="feminino" value="F"> <label for="feminino">Feminino</label> </fieldset> </form>

Dessa forma, o radio ou botão de rádio nada mais é que um tipo de campo que já contém informações predefinidas e que o usuário deverá marcar... Tendo a opção de selecionar nenhuma ou uma caixa por grupo (somente o valor da caixa marcada que será enviado para o back-end).

If you ganna use input file, always use METHOD POST.