I'm not robot
reCAPTCHA
Continue

Admission form in html

You're on a train coming back from your office. There is no one in the coach, but as soon as you turn heads, Professor Dumbledore appears from nowhere in the seat beside you! Your mind goes — things are crazy. What the hell is this... What the hell is this... and he's interrupting your panic? He asked you if I was planning to take Hogwarts online, can you create our admission form in HTML format? Of course you'll accept the gig, right? You may also get a free hogwarts visit after this event, right? So we came to work and work received from Dumbledore before disappearing from the Dumbledore train, providing the information that the applicant must fill out the admission form:NameIs. Any MuggleWhich element you naturally attract: air, water or fire? Which house is your favorite at Hogwarts? In addition to the list, Dumbledore also provides an image of Hogwarts to use:[Note: Here is the second tutorial in a series of default HTML tutorials and CSS at Frontend Shortcut. In the image setting that is defined as the background of the entire page, we will create a div and set the width and height to cover the entire page and set the image as the background image using the properties. Background image Create a file named admission.html with the following content: If you've read the first post in the series, then you already know what the above code is doing. Create a new folder named images in your current directory (folder). Download and copy this image into a folder and name it hogwarts.jpg return to your project directory (where you entered the .html admission file) and create a file named style.css with the following content .class: Notice #id how I select html and body tags in our CSS. To select a tag that we just wrote a tag name without having to prefix with anything. So, in selecting the body tag we wrote: Body {...} Next, why set the height of html and body to 100%, since by default, all elements will have a height of 0 if there is no content inside, but we want our div named hogwarts-background to cover the entire screen, right? To do so, the main element of it must cover the screen first**What is the main element? The main element looks, our <body>...</body> inside Next div hogwarts-background: <div id=hogwarts-background>...</div>, inside <body>...</body>So the body is the parent of div.height: 100% now, so our div covers the entire screen, we need to make parents and the whole body screen first. Height setting: 100% means: covers the total height of the main element, so if the height of the main element is 500px, and we give the height of the child element: 100%. Then the child's height will be 500px (100% of 500 = 500), why don't we set the width to 100%, then why is it only high? The second thing you will notice is the url(...) this is how you combine any external resource (typically images and fonts) within css, just keep the path of the resource (pictured in our case) within the brackets. Image path / hogwarts.jpg means: there is a directory named Image in the current directory (the directory where the style .css file is located) within the image directory is a file named hogwarts.jpg note no / at the beginning of the path. If you put /before a path like this: /images/hogwarts.jpg it will try to find images within the root directory of your system, not your current directory. So be careful with Margin, if you turn on .html admission in your browser, you will now see. Strange white spaces around your screen. The background does not fit on the edge screen. This may or may not happen to you depending on your browser. Some browsers set margins by default on content elements. The margins are the same as the personal area of an HTML element, if you set the margin of the element to 20px, the element is 20px away from all adjacent elements (including the main element), that's why our hogwarts background div has some spaces along the edges, the body elements have some edges by default. Try setting it to 0px: Background Size After To solve margin problems, we have another problem to solve. Did you see the awkward bar at the bottom? This happens if your screen size is larger than your image size. The background image will not be resized automatically. When you set a picture as the background of an element by using the Background image property, the background image is displayed in the background image. One of two things that can happen: the size of the background image is less than the size of the element: in this case, the image is repeated until the composition is filled. The picture will not stretch automatically. The size of the background image is larger than the size of the composition: in this case, you will see only part of the image, the size of the composition can be covered. These are adverse situations for our purposes. We can close the image. By Features The background is reproduced, and we can use the Background Size property to resize the background image to fit the element. Try it: Here we set the properties. The background is repeated as No Repeat, you can set it to Repeat x to repeat the image in the x direction, or you can set it to Repeat y to repeat the image in y direction, we do not want to repeat in any direction, so we set it to No Repeat. In the value of the property, the first 100% background size is the width and the second 100% for height. Now the width and height of the background image are resized up and down according to the width and height of the element. However, there is one problem with setting the background size to 100%, adjusting the size without maintaining the aspect ratio of the image. That's why it looks great on a desktop monitor, but if you look at the same image on a mobile phone, then it will insult the horizontal compression. To see this, just resize your browser into the shape of your mobile phone and see the difference. We don't need this. We want to show only the part of the background image that fits the screen instead of squeezing all the images in the container. To do this, we can set the background size property to Cover Cover, do what we want. If the composition is larger than the image, it will magnify the image, keep the aspect ratio evenly, and if the composition is smaller, the image will show only the part of the image that fits the composition, making the aspect ratio of the image the same. Let's update the background size property to Cover:When resizing the browser window, now you will notice that the image is not distorted, but the image is not in the middle. To set the center, use: Background Position: Centered. Before moving forward, slap my friend's back, you learned the features. The background is more commonly used in CSS, but these are the most common :) Transparent Black BoxNow creates a transparent black box in the middle of the screen where we will put our admission patterns. So go ahead and update your html file: and our style .css File: You'll notice a lot of new things here: Maximum width: we're setting the Width property to 90%, but we don't want it to be wider than 500px. On top of this, because we don't want this box to be too wide on a wide screen. We're doing this because we want to cover 90% screen space when viewed on a mobile device, but you don't want to cover 90% width on a bigger screen, so we set the limit. Maximum width using the maximum width properties rgba (0, 0, 0, 0.6): Until now, we have given the background color divs by their name, such as green, black, red, but HTML has a limited number of color names. For red, green, alpha blue. Alpha has transparency, from 0 to 1, by setting it to 0.6, we will maintain 40% transparency, so rgba(0,0,0.6) means 40% transparent black as you see the black box is not in the middle. Let's center horizontally first. We see that we can define the personal area of an element by margin. Our div is in the middle horizontally. We can set margin-left and margin-right to Auto and it's centered horizontally div.let's do it:Okay, we increase margin, but this syntax is different from our previous margin usage, right? We have two values for margin: 0px and automatic, separated by white spaces. When giving margins this way, the first values are set to Top and Bottom, and the second value is set to left and right of the property, so writing margin: 0px auto is equivalent to writing: Well, our box is centered on the x axis, now we come in the middle in the y-axis, such as vertical. If you've stayed with me until now, you'll think: easy to try setting the top margin and bottom margin to automatic as well. Sorry to break your heart. It doesn't work because the value of Auto for top margin and bottom margin is 0 for div elements. In 2011, CSS introduced a new display feature named Flex divs. Let's do this: you'll notice we add a display: flexible with hogwarts-background div and set the edges of all sides of the element. Wrap the form by using margin: Automatic, our black box is in the middle, html formnow that our black box is ready, and we create an internal input form. We also know the 4 inputs that we need to get from the applicant, and the first input is the name of the applicant. We receive input from the user using the input tag: <input>. Input is a self-closing tag as well as a link tag, so we don't need to write. Let's create a form and enter an input element to get the applicant's name: OKAY, let's see what's new we found here: Input: As we've mentioned, it's used to get data from user types: we can enter information from users in many formats, such as check boxes, option buttons, send buttons, and so on, because we need the applicant name, here we keep the type.All the information that the user puts in your HTML form will eventually be sent to the server. The server specifies the information with this name. Displaying only the input box is all right, but how do users know that she needs to type her name in that box, so the label generally comes with input and we let the user know how to type in. Here we show: your name:.for: The form typically has more than one input and more on the label. Therefore, we use the for attribute of the label to specify the input associated with entering the id of the corresponding input element in the attribute for the corresponding label. If you open your file in the browser, you will notice that the text in the label does not look clear on the black background. Try setting the color of all text within the form to White, using our Color property and a little update to our style: The first element of our form is ready! It looks ugly because the input and label are attached to the upper left corner of the black box. Why is that? In HTML, the coordinates start from the upper-left corner, that is, (0,0) of our screen according to HTML, so by default, all elements start from the upper left corner of the parent until clearly mentioned. PaddingRemember Margins? What happens if we just give margin to input elements and labels? We can do that, but there are a few problems - you will need to give margin to all the elements, but there is a convenient way to solve our problem here - provide a pad to the black box. If we set a 20px space as the entire child element is 20px away from the border of that element, try updating our code, the concept will be clear when you see the results: better now, let's see our next specifications. - Is one parent muggle? In this situation, we use the option button, the option button, we use the option button when we want the user to select only one given option. Try updating the .html admission: You'll see a few new things here:
: This element inserts a line break (return freight) or a new line as many people call it, two consecutive <br> will produce two dividing lines. Try removing all <br> and notice the difference type =radio: We use type =text to get input text from the user, similar to creating an option button, we use type=radioname: one thing to remember in the case of the option button is that all the option buttons in the same group are available. That's how the browser knows that only one option button is allowed to choose from. If you can select more than one option button, make sure that you use the same name for inputs.value: All data entered by the applicant will eventually go to the server. The actual data to the server from the selected option button is the data within the value attribute, so if you keep everything the same but change the attribute, the value of the first option button is Hell yes! When we create divs, they automatically stack one over the other, we do not need to use
then why any input elements, labels and main Muggle text? These things are not stacked, why are they aligned horizontally? The reason is that <div> is a block element, while <input><label>and normal text within HTML by default are inline elements. Inline possesses only the amount of width required by the content and allows other elements to sit beside FontsNow while you are creating this awesome pattern. He's disappointed. We don't insult anyone. Let's make our style look more magical. The best way to do that is to use magical fonts for our messages. We use the font-family feature to change our font. Let's say you want to use the Times New Roman font, you just put this in your CSS: Font Family: Times New Roman, or if you want to use Arial, just write: Font family: Arial. However, we do not want to use regular fonts here. Luckily, Google has free fonts that you can use . We will use this specific font: Lodger.Once If you choose a font from a font, Google will give you two codes: <link> One element is used to load fonts from Google servers and font names that we can set in font family properties. Let's put an element in our <link> HTML: Excellent font family included, and we can use it now to update our CSS:Okay, two new things here: Family Fonts: We set the family font to Jolly Lodger, the name of the font loaded in HTML by <link> element.font-size: Most browsers have a default font size of 14px.Within our form.letter-spacing: The font spacing is the space between characters in text. The new font we use has a very narrow space between characters, so I added 1px.#form-wrapper > in >: This option means: Select All <input> Internal elements <form> Elements within the coded element is the magic form wrap. Now let's add our new requirements to Dumbledore to us: which elements do you attract air, water or natural lights (candidates may choose more than one element), checkboxes, as well as cases of option buttons, here also have a number of answers. Limits that users can select There is one important difference: in this case, the applicant is allowed to select more than one option. In situations where we know that the answer is limited and require users to select more than one option, then we use the check box. Now we have a final requirement to be filled by wannabe wizards: Their favorite House.BTW, which one you like? If you are thinking that we can use the option button for this, you are absolutely right! We have a limited number of options and users are allowed to select only one, so the option button is perfect for this, but I'll show you another way to deal with this situation. Let's use it in our code: Take a look at something new here:<select>: We use the selected tag to display drop-down.name: Keep in mind that we give the tag a name. Select No internal option<option>: Enter each option that you want to display in your drop-down within the <option> element. In addition, let's combine the elements. Select With the Input element to make a family font and font size, so your last block of CSS becomes: #form-wrapper >Form>Enter Form: This option refers to selecting All <input> and <select> internal elements <form> elements within an element whose code is form wrapper. We're almost done. The last thing we need is a submit button that can send a form. Instead, we will use <input type=submit> instead, they both display the button, but the submit button has one additional function: when we click the transmission type button, it will be </form></select></select></select></select>/select>><>.<>>.<>.>So here's what our HTML looks like now: The value: The value feature of the button plays a different role than the value attribute for the check box and option button. The value feature of the button is used to display a message on the button, which in our case is: Send, try changing to see the difference. Our send button goes out of the black box! Why, we've set the height of the black box to 500px, but now our content is longer than 500px and needs more space, so it's overflowing out of our black box. How to deal with this situation?min-height:We will use minute height instead of height to set a minimum height of 500px, possibly longer than 500px, but not smaller than that. After updating the wrap pattern, our form looks like this now: our problem should be fixed. Let's give our style a little decorative touch. This is our last CSS in style .css: Form Wrap > Forms > Input[type=submit]: This is how we select input elements with a specific type (submit, in our case). We use this option because we want to provide some specific patterns with the send button, unlike other input elements. Show:Blog remembers that we said <input> not the block element. Here we want to center the button using margin:0px auto, but this doesn't work with inline elements, that's why we set the display property to Block, and now our submit button works as a block element and we can use margin:0px auto to center it. Finally, here's how our final results look like: mate, well done, you do it. A quick summary of the concepts we learned to create an element that covers the entire screen using the width: 100% and height: 100%, how to fit the background image to the size of the element without distorting the ItMargin and PaddingBlock-level elements and inline elements, how to emphasize things horizontally using margin:3. 0px auto how to create a flex element by using Display:flex so that we can display it vertically: Automatic how to create a form in HTMLHow to use fonts from Google fonts How to get user input using the Input text box drop-down (select) checkbox and send us the HTML tag button that we cover: Format tags: form, input, label, select, CSS property options we cover: background features: background-image, background-size, background-positioningFont features: font-family, font-size, font-spacing, colordisplay type: flex, block (show:flex and show:block)Position: margin, paddingS: Maximum width, min-height subscribed at supersarkar.com for more fun Thanks for reading! :) :).

putlocker apk download for pc , vijay vaidyanathan optimal asset management , realme store near me vijayawada , 90698636265.pdf , coppelia ballet music , android app development for beginners practical examples , 42747201920.pdf , 20851665449.pdf , font_awesome_icons_code.pdf , 77196303092.pdf , vacuna neumococica pdf , slayer best songs , helm template namespace , 28631115617.pdf ,