1. basic concepts of git
   1. (1) how to clone the link from git hub:
   2. open visual studio code -> new window -> click explore -> clone repository -> paster the link -> enter
   3. (2) how to submit the file to git hub:
   4. Git add . -> git commit -m “statement” -> git push
   5. (3) Why do we use git?
   6. To record different versions of code, share with other people to look.
2. Summarize all the abstracts (tags) of HTML. (including how to use)

HTML format:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Delightful Ducks</title>

<link rel="stylesheet" href="./duck.css">

</head>

<body>

</body>

</html>

<h1><h2><h3><h4><h5><h6> </h6></h5></h4></h3></h2></h1>

<p> </p>

<a href=”<https://www.google.com>”>This is a link</a>

<img src=”image.jpg” alt=”image”>

<br> change line

<hr> have a line between two column

The HTML <pre> element defines preformatted text.

HTML Formating

* <b> - Bold text
* <strong> - Important text
* <i> - Italic text
* <em> - Emphasized text
* <mark> - Marked text
* <small> - Smaller text
* <del> - Deleted text
* <ins> - Inserted text
* <sub> - Subscript text
* <sup> - Superscript text

The HTML <q> tag defines a short quotation.

Use an Image as a Link

<a href="default.asp">  
<img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;">  
</a>

Use a button as a Link

<button onclick="document.location='default.asp'">HTML Tutorial</button>

Link bookmark

<h2 id="C4">Chapter 4</h2>

<a href="#C4">Jump to Chapter 4</a>

<a href="html\_demo.html#C4">Jump to Chapter 4</a>

Favicon

<link rel="icon" type="image/x-icon" href="/images/favicon.ico">

Table

First version

<table>  
  <tr>  
    <th>Company</th>  
    <th>Contact</th>  
    <th>Country</th>  
  </tr>  
  <tr>  
    <td>Alfreds Futterkiste</td>  
    <td>Maria Anders</td>  
    <td>Germany</td>  
  </tr>  
  <tr>  
    <td>Centro comercial Moctezuma</td>  
    <td>Francisco Chang</td>  
    <td>Mexico</td>  
  </tr>  
</table>

Second version

<table>

<caption> A complex table</caption>

<thead>

<tr>

<th colspan="2"> Invoice #987654321</th>

<th colspan="2"> 31st Feburary, 2019</th>

</tr>

<tr>

<td colspan="2">

<h3 class="text-left"> Pay to:</h3>

<p class="text-left">Buy &amp; Large 1td.</p>

<p class="text-left">123 Some St.</p>

<p class="text-left">Nowherevile, NZ 9999</p>

</td>

<td colspan="2">

<h3 class="text-left"> Customer:</h3>

<p class="text-left"> Walther White</p>

<p class="text-left"> 308 Negra Arroyo Lane</p>

<p class="text-left"> Albuquerque, NM 87104</p>

</td>

</tr>

</thead>

<tbody>

<tr>

<th> Name / description</th>

<th> Qty.</th>

<th> @</th>

<th> Cost</th>

</tr>

<tr>

<td class="text-left">Test tubes</td>

<td>30</td>

<td>2.50</td>

<td>75.00</td>

</tr>

<tr>

<td class="text-left">Flasks</td>

<td>10</td>

<td>5.00</td>

<td>50.00</td>

</tr>

</tbody>

<tfoot>

<tr>

<th colspan = "3"> <strong>Subtotal </strong></th>

<td>125.00</td>

</tr>

<tr>

<th colspan = "2"> <strong>Tax </strong></th>

<td>8%</td>

<td>10.00</td>

</tr>

<tr>

<th colspan = "3"> <strong>Grand Total </strong></th>

<td>$135.00</tdgi>

</tr>

</tfoot>

</table>

HTML list

Unordered list

<ul>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ul>

Ordered list

<ol>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>

Description list

<dl>  
  <dt>Coffee</dt>  
  <dd>- black hot drink</dd>  
  <dt>Milk</dt>  
  <dd>- white cold drink</dd>  
</dl>

HTML Form

<form action="https://echo.trex-sandwich.com" method = "post">

<fieldset>

<legend> Your details</legend>

<p> Name: <br /> <input type = "text" name = "name" size="15" maxlength="30" /></p>

<p> Email: <br /><input type = "text" name = "name" size="15" maxlength="30" /></p>

<p> Your city: <br /><input type = "text" name = "name" size="15" maxlength="30" /></p>

</fieldset>

<fieldset>

<legend>Questionnaire</legend>

<p> How are you finding this course? Tick all applicable <br />

<input type = "checkbox" name="finding" value= "Enjoying it" checked="checked"/>Enjoying it

<input type = "checkbox" name="finding" value= "Challenging" />Challenging

<input type = "checkbox" name="finding" value= "Still catching up" />Still catching up

<input type = "checkbox" name="finding" value= "Too hard" />Too hard

<input type = "checkbox" name="finding" value= "Too many exercise" />Too many exercise

<input type = "checkbox" name="finding" value= "Not enough exercise" />Not enough exercise

<input type = "checkbox" name="finding" value= "Confused" />Confused

</p> <br />

<p>

Choose the one that best describes your experience so far <br />

<input type = "radio" name="experience" value= "Learning a lot" />Learning a lot

<input type = "radio" name="experience" value= "Not learning enough" />Not learning enough

<input type = "radio" name="experience" value= "Okay" />Okay

</p>

<p>

pick your favorite animal:

<select name="animal">

<option>Mandarin duck</option>

<option>Quokka</option>

<option>Wombat</option>

<option>Lynx</option>

<option>Pika</option>

<option>Red Panda</option>

</select>

</p>

<p>

<textarea id="profile\_text" name = "profile\_text\_area" rows = "4" cols = "40" placeholder="Additional comments here"> </textarea>

</p>

</fieldset>

<button><input type="submit" value="Done!"> </button></a>

<button> <input type="reset" value="Clear!" /></button>

</form>

* <input type="button">
* <input type="checkbox">
* <input type="color">
* <input type="date">
* <input type="datetime-local">
* <input type="email">
* <input type="file">
* <input type="hidden">
* <input type="image">
* <input type="month">
* <input type="number">
* <input type="password">
* <input type="radio">
* <input type="range">
* <input type="reset">
* <input type="search">
* <input type="submit">
* <input type="tel">
* <input type="text">
* <input type="time">
* <input type="url">
* <input type="week">
* <input>
* <label>
* <select>
* <textarea>
* <button>
* <fieldset>
* <legend>
* <datalist>
* <output>
* <option>
* <optgroup>

1. Summarize all the abstracts (tags) of CSS. (including how to use)

The CSS rule below will be applied to the HTML element with id="para1":

#para1 {  
  text-align: center;  
  color: red;  
}

In this example all HTML elements with class="center" will be red and center-aligned:

.center {  
  text-align: center;  
  color: red;  
}

In this example only <p> elements with class="center" will be red and center-aligned:

p.center {  
  text-align: center;  
  color: red;  
}

In this example the <p> element will be styled according to class="center" and to class="large":

<p class="center large">This paragraph refers to two classes.</p>

The universal selector (\*) selects all HTML elements on the page.

\* {  
  text-align: center;  
  color: blue;  
}

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

CSS background

Background-color - The background-color property specifies the background color of an element.

The opacity property specifies the opacity/transparency of an element. It can take a value from 0.0 - 1.0. The lower value, the more transparent:

Background-image - The background-image property specifies an image to use as the background of an element.

By default, the image is repeated so it covers the entire element.

body {  
  background-image: url("paper.gif");  
}

Background-repeat -If the image above is repeated only horizontally (background-repeat: repeat-x;), the background will look better:

Repeat-x, Repeat-y, no-repeat

Background-attachment - The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page):

Background-position - The background-position property is used to specify the position of the background image.

background

CSS border

The border-style property specifies what kind of border to display.

The following values are allowed:

* dotted - Defines a dotted border
* dashed - Defines a dashed border
* solid - Defines a solid border
* double - Defines a double border
* groove - Defines a 3D grooved border. The effect depends on the border-color value
* ridge - Defines a 3D ridged border. The effect depends on the border-color value
* inset - Defines a 3D inset border. The effect depends on the border-color value
* outset - Defines a 3D outset border. The effect depends on the border-color value
* none - Defines no border
* hidden - Defines a hidden border

p.dotted {border-style: dotted;}  
p.dashed {border-style: dashed;}  
p.solid {border-style: solid;}  
p.double {border-style: double;}  
p.groove {border-style: groove;}  
p.ridge {border-style: ridge;}  
p.inset {border-style: inset;}  
p.outset {border-style: outset;}  
p.none {border-style: none;}  
p.hidden {border-style: hidden;}  
p.mix {border-style: dotted dashed solid double;}

border-width: 2px

border-color: red

border-radius: 5px

CSS margin: The CSS margin properties are used to create space around elements, outside of any defined borders.

CSS has properties for specifying the margin for each side of an element:

* margin-top
* margin-right
* margin-bottom
* margin-left

All the margin properties can have the following values:

* auto - the browser calculates the margin
* length - specifies a margin in px, pt, cm, etc.
* % - specifies a margin in % of the width of the containing element
* inherit - specifies that the margin should be inherited from the parent element
* **margin: 25px 50px 75px 100px;**
  + top margin is 25px
  + right margin is 50px
  + bottom margin is 75px
  + left margin is 100px

## **Margin Collapse**

Top and bottom margins of elements are sometimes collapsed into a single margin that is equal to the largest of the two margins.

This does not happen on left and right margins! Only top and bottom margins!

Look at the following example:

CSS padding

CSS has properties for specifying the padding for each side of an element:

* padding-top
* padding-right
* padding-bottom
* padding-left

All the padding properties can have the following values:

* length - specifies a padding in px, pt, cm, etc.
* % - specifies a padding in % of the width of the containing element
* inherit - specifies that the padding should be inherited from the parent element

CSS height and width:

The height and width properties may have the following values:

* auto - This is default. The browser calculates the height and width
* length - Defines the height/width in px, cm, etc.
* % - Defines the height/width in percent of the containing block
* initial - Sets the height/width to its default value
* inherit - The height/width will be inherited from its parent value
* The max-width property is used to set the maximum width of an element.
* The max-width can be specified in length values, like px, cm, etc., or in percent (%) of the containing block, or set to none (this is default. Means that there is no maximum width).
* The problem with the <div> above occurs when the browser window is smaller than the width of the element (500px). The browser then adds a horizontal scrollbar to the page.
* In this situation, using max-width will improve the browser's handling of small windows.

If you for some reason use both the width property and the max-width property on the same element, and the value of the width property is larger than the max-width property; the max-width property will be used (and the width property will be ignored).

CSS text/font

The color property is used to set the color of the text. The color is specified by:

* a color name - like "red"
* a HEX value - like "#ff0000"
* an RGB value - like "rgb(255,0,0)"
* text-align:
  + The text-align property is used to set the horizontal alignment of a text.
  + A text can be left or right aligned, centered, or justified.
  + The following example shows center aligned, and left and right aligned text (left alignment is default if text direction is left-to-right, and right alignment is default if text direction is right-to-left):

Text-align include: left, right, center, justify

* text-align-last- The text-align-last property specifies how to align the last line of a text.
* direction
* unicode-bidi
* vertical-align - The vertical-align property sets the vertical alignment of an element.
* img.a {  
    vertical-align: baseline;  
  }  
    
  img.b {  
    vertical-align: text-top;  
  }  
    
  img.c {  
    vertical-align: text-bottom;  
  }  
    
  img.d {  
    vertical-align: sub;  
  }  
    
  img.e {  
    vertical-align: super;  
  }

## **Text Decoration**

In this chapter you will learn about the following properties:

* text-decoration-line
* text-decoration-color
* text-decoration-style
* text-decoration-thickness
* text-decoration

The text-decoration-line property is used to add a decoration line to text.

h1 {  
  text-decoration-line: overline;  
}  
  
h2 {  
  text-decoration-line: line-through;  
}  
  
h3 {  
  text-decoration-line: underline;  
}  
  
p {  
  text-decoration-line: overline underline;}

h1 {  
  text-decoration-line: overline;  
  text-decoration-color: red;  
}  
  
h2 {  
  text-decoration-line: line-through;  
  text-decoration-color: blue;  
}  
  
h3 {  
  text-decoration-line: underline;  
  text-decoration-color: green;  
}  
  
p {  
  text-decoration-line: overline underline;  
  text-decoration-color: purple;  
}

h1 {  
  text-decoration-line: underline;  
  text-decoration-style: solid;  
}  
  
h2 {  
  text-decoration-line: underline;  
  text-decoration-style: double;  
}  
  
h3 {  
  text-decoration-line: underline;  
  text-decoration-style: dotted;  
}  
  
p.ex1 {  
  text-decoration-line: underline;  
  text-decoration-style: dashed;  
}  
  
p.ex2 {  
  text-decoration-line: underline;  
  text-decoration-style: wavy;  
}  
  
p.ex3 {  
  text-decoration-line: underline;  
  text-decoration-color: red;  
  text-decoration-style: wavy;  
}

text-decoration-thickness:5px;

text-transform: uppercase / lowercase/capitalize

text-shadow: The text-shadow property adds shadow to text.

In its simplest use, you only specify the horizontal shadow (2px) and the vertical shadow (2px):

h1 {  
  text-shadow: 2px 2px;  
}

Then, add a blur effect (5px) to the shadow:

## **Text shadow effect!**

h1 {  
  text-shadow: 2px 2px 5px red;  
}

font:

font-family: / font-style: / font-weight: / font-variant

font-size: /

CSS link:

* a:link - a normal, unvisited link
* a:visited - a link the user has visited
* a:hover - a link when the user mouses over it
* a:active - a link the moment it is clicked

CSS list:

ul.a {  
  list-style-type: circle;  
}  
  
ul.b {  
  list-style-type: square;  
}  
  
ol.c {  
  list-style-type: upper-roman;  
}  
  
ol.d {  
  list-style-type: lower-alpha;  
}

CSS table:

Border: 1px solid;

border-collapse: collapse;

width: / height:

The text-align property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>.

By default, the content of <th> elements are center-aligned and the content of <td> elements are left-aligned.

To center-align the content of  <td> elements as well, use text-align: center:

For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:

tr:nth-child(even) {background-color: #f2f2f2;}

CSS Display:

display: none; is commonly used with JavaScript to hide and show elements without deleting and recreating them. Take a look at our last example on this page if you want to know how this can be achieved.

The <script> element uses display: none; as default.

Setting the display property of an element only changes **how the element is displayed**, NOT what kind of element it is. So, an inline element with display: block; is not allowed to have other block elements inside it.

visibility:hidden; also hides an element.

However, the element will still take up the same space as before. The element will be hidden, but still affect the layout:

Display: inline / display: block / display: inline-block

Common values

- Inline: sits next to other elements

- takes up “just enough” width and height

- Block: forces line break

- default: take up all horizontal width and “just enough” height

- rule can set width and height

- Inline-block

- Same as inline, but accepts height and width

1. None

## **The float Property**

The float property is used for positioning and formatting content e.g. let an image float left to the text in a container.

The float property can have one of the following values:

* left - The element floats to the left of its container
* right - The element floats to the right of its container
* none - The element does not float (will be displayed just where it occurs in the text). This is default
* inherit - The element inherits the float value of its parent

In its simplest use, the float property can be used to wrap text around images.

When we use the float property, and we want the next element below (not on right or left), we will have to use the clear property.

The clear property specifies what should happen with the element that is next to a floating element.

The clear property can have one of the following values:

* none - The element is not pushed below left or right floated elements. This is default
* left - The element is pushed below left floated elements
* right - The element is pushed below right floated elements
* both - The element is pushed below both left and right floated elements
* inherit - The element inherits the clear value from its parent

When clearing floats, you should match the clear to the float: If an element is floated to the left, then you should clear to the left. Your floated element will continue to float, but the cleared element will appear below it on the web page.

1. basic concepts of HTML
   1. (1) including what kind of things should be put into <head>
   2. <meta>, <title>, <link>,<favicon>
   3. (2) including what kind of things should be put into <body>
   4. Heading, paragraph, table, list, image, form, class, ID
   5. (3) what’s the difference between “id” and “class.”
   6. ID only use once. Can help to jump the content
   7. Class can group use, easy to control several stuff.
   8. (4) why do we need to use <div> in HTML
   9. We need to make a block on our page.
   10. (5) how do we use the image as the link
   11. <a><img></a>
   12. (6) what are the key points in HTML file.

<! DOCTYPE html>

* 1. <head></head><body></body>

1. Must have close mark!!!
2. Readability & Maintainability
3. Accessibility (a11y)
4. Performance & Optimization
5. Responsiveness & Mobile-Friendliness
6. Cross-Browser Compatibility
   1. (7) how can we connect with other HTML file or CSS file.

Use link tag

1. basic concepts of CSS
   1. (1) what’s the difference between p, .p, #p, div p
   2. P => paragraph CSS setting
   3. .p => class p CSS setting
   4. #p => id p CSS setting
   5. Div p => p CSS setting (the child of div)
   6. (2) What’s the difference between margin and padding?
   7. Margin is outside the border
   8. Padding is between the border and content.
   9. (3) Introduce three different ways to define the color.
   10. Text, rgb(red, green, blue, ALPHA), Hex(#rrggbb)
   11. (4) how can we interact with images
   12. (5) explain “document object model”