**HTML Responsive Web Design**

Last Updated : 30 Jul, 2024

HTML Responsive Web Design is a modern approach to web development that ensures web pages look great on any device, from phones and tablets to desktop computers. It focuses on making HTML elements automatically adjust—resizing, hiding, or repositioning based on the screen size. This approach guarantees a seamless and visually appealing user experience across various devices.

In this article, we’ll explore practical examples and techniques for creating HTML responsive web designs that adapt perfectly to different screen sizes, ensuring optimal functionality.

**What is Responsive Web Design?**

Responsive Web Design is an approach to web development that ensures web pages render well on a variety of devices and window or screen sizes. It involves using fluid grids, flexible images, and CSS media queries to adapt the layout of a website dynamically based on the screen size, orientation, and resolution of the device being used.

This technique allows websites to provide an optimal viewing experience, easy reading, and navigation with minimal resizing, panning, and scrolling across different devices, from desktop monitors to mobile phones.

**Examples of HTML Responsive Web Design**

**Table of Content**

* [1. HTML Viewport meta tag for Responsive Web Design](https://www.geeksforgeeks.org/html-responsive-web-design/#1-html-viewport-meta-tag-for-responsive-web-design)
* [2. Responsive Images](https://www.geeksforgeeks.org/html-responsive-web-design/#2-responsive-images)
* [3. Responsive Texts](https://www.geeksforgeeks.org/html-responsive-web-design/#3-responsive-texts)
* [4. CSS media Queries](https://www.geeksforgeeks.org/html-responsive-web-design/#4-css-media-queries)
* [5. Responsive Layouts](https://www.geeksforgeeks.org/html-responsive-web-design/#5-responsive-layouts)

**1. HTML Viewport meta tag for Responsive Web Design**

The [HTML viewport](https://www.geeksforgeeks.org/html-viewport-meta-tag-for-responsive-web-design/) is the visible area of the screen that users see. It changes depending on the device being used. With this approach, we set the width of web pages to match the available screen width, making it 100%. This helps content adapt and look good on any device, ensuring a responsive layout.

**Syntax**

<meta name="viewport" content= "width=device-width, initial-scale=1.0">

**Example:**Use of the HTML viewport meta tag for responsive pages.



1

<!DOCTYPE html>

2

<html>

3

​

4

<head>

5

<title>GeeksforGeeks</title>

6

<meta charset="utf-8"

7

name="viewport"

8

content="width=device-width,

9

initial-scale=1.0" />

10

<style>

11

.gfg {

12

font-size: 40px;

13

font-weight: bold;

14

color: green;

15

text-align: center;

16

}

17

​

18

.geeks {

19

font-size: 17px;

20

text-align: center;

21

}

22

​

23

p {

24

text-align: justify;

25

}

26

</style>

27

</head>

28

​

29

<body>

30

<div class="gfg">GeeksforGeeks</div>

31

<div class="geeks">HTML Introduction</div>

32

​

33

<p>

34

HTML stands for HyperText Markup Language. It is

35

used to design web pages using a markup

36

language. HTML is a combination of Hypertext and

37

Markup language. Hypertext defines the link

38

between web pages. A markup language is used to

39

define the text document within the tag which

40

defines the structure of web pages. This

41

language is used to annotate (make notes for the

42

computer) text so that a machine can understand

43

it and manipulate text accordingly. Most markup

44

languages (e.g. HTML) are human-readable. The

45

language uses tags to define what manipulation

46

has to be done on the text.

47

</p>

48

</body>

49

​

50

</html>

**Output**

Video Player

00:00

00:06

**2. Responsive Images**

Responsive images play a key role in responsive websites. These are images that can adjust their size, getting bigger or smaller, based on the width of the browser. By being responsive, images enhance user experience across different devices with varying screen sizes. The following are the techniques to use the responsive images:

**2.1 Using width Property**

The image can be responsive & scale up & down with the help of CSS width property by setting its value as 100%.

**Syntax**

<img src="..." style="width:100%;">

**Example:** In this example, we will use the image width property to occupy 100% of the screen width.



1

<!DOCTYPE html>

2

<html>

3

​

4

<head>

5

<meta name="viewport" content=

6

"width=device-width, initial-scale=1.0" />

7

</head>

8

​

9

<body>

10

<img class=".img-fluid" src=

11

"https://media.geeksforgeeks.org/wp-content/uploads/20220201191443/logo-200x32.png"

12

style="width: 100%" />

13

​

14

<h2>Responsive Images</h2>

15

16

<p>

17

Responsive images are just a part of Responsive

18

websites. Images that can change their

19

dimensions, scaling them up or down, according

20

to the browser width are responsive images. The

21

above image is responsive as it is adjusting

22

itself according to the width of the browser.

23

</p>

24

</body>

25

​

26

</html>

**Output**

Video Player

00:00

00:08

**2.2 Using the max-width Property**

The max-width property sets the maximum width of an element. It restricts the element’s width from exceeding a certain value. If the content is larger than this maximum width, it wraps to the next line. However, if the content is smaller, the property has no impact. Also, if the content is larger than its original size, it won’t scale up beyond the specified max-width.

**Syntax:**

<img src="..." style="max-width:100%; height:auto;">

**Example: Implementation of the max-width Property for making the Responsive Images**



1

<!DOCTYPE html>

2

<html>

3

​

4

<head>

5

<meta name="viewport"

6

content="width=device-width,

7

initial-scale=1.0" />

8

</head>

9

​

10

<body>

11

<img class=".img-fluid"

12

src=

13

"https://media.geeksforgeeks.org/wp-content/uploads/20220201191443/logo-200x32.png"

14

style="max-width:100%;

15

height:auto;" />

16

<h2>Responsive Images</h2>

17

<p>

18

Responsive images are just a part of Responsive

19

websites. Images that can change their

20

dimensions, scaling them up or down, according

21

to the browser width are responsive images. The

22

above image is responsive as it is adjusting

23

itself according to the width of the browser.

24

</p>

25

</body>

26

​

27

</html>

**Output**

Video Player

00:00

00:11

**2.3 Responsive Image for different Browser Width**

To make images responsive, HTML provides the <picture> element. It gives web developers the flexibility to specify different image resources depending on the size of the browser window

**Example**



1

<!DOCTYPE html>

2

<html>

3

​

4

<head>

5

<title>HTML Responsive Web Design</title>

6

</head>

7

​

8

<body style="text-align: center;">

9

<h1 style="color: green;">GeeksforGeeks</h1>

10

​

11

<h2>HTML picture Tag</h2>

12

​

13

<picture>

14

<source media="(min-width: 700px)"

15

srcset=

16

"https://media.geeksforgeeks.org/wp-content/uploads/20190825000042/geeks-221.png">

17

​

18

<source media="(min-width: 450px)"

19

srcset=

20

"https://media.geeksforgeeks.org/wp-content/uploads/20190802021607/geeks14.png">

21

​

22

<img src=

23

"https://media.geeksforgeeks.org/wp-content/uploads/20190808102629/geeks15.png"

24

alt="GFG">

25

</picture>

26

</body>

27

​

28

</html>

**Output**

Video Player

00:00

00:11

**3. Responsive Texts**

In this method, we set font sizes using %, vw, vh, etc. This ensures that text sizes are responsive, adjusting automatically until reaching a certain limit. Once the limit is reached, the content is justified to fit within the available width.

**Example: This example demonstrates Responsive Web Design by making Responsive Texts.**



1

<!DOCTYPE html>

2

<html>

3

​

4

<head>

5

<style>

6

body {

7

max-width: 100%;

8

}

9

​

10

.gfg {

11

font-size: 7vw;

12

font-weight: bold;

13

color: green;

14

text-align: center;

15

}

16

​

17

.geeks {

18

font-size: 5vw;

19

text-align: center;

20

}

21

​

22

p {

23

font-size: 3vw;

24

text-align: justify;

25

}

26

</style>

27

</head>

28

​

29

<body>

30

<div class="gfg">GeeksforGeeks</div>

31

<div class="geeks">HTML Introduction</div>

32

​

33

<p>

34

HTML stands for HyperText Markup Language. It is

35

used to design web pages using a markup

36

language. HTML is a combination of Hypertext and

37

Markup language. Hypertext defines the link

38

between web pages. A markup language is used to

39

define the text document within the tag which

40

defines the structure of web pages. This

41

language is used to annotate (make notes for the

42

computer) text so that a machine can understand

43

it and manipulate text accordingly. Most markup

44

languages (e.g. HTML) are human-readable. The

45

language uses tags to define what manipulation

46

has to be done on the text.

47

</p>

48

</body>

49

​

50

</html>

**Output:**

Video Player

00:00

00:08

**4. CSS media Queries**

The Media query in CSS is essential for crafting responsive web designs. It ensures that web pages adapt to various screen sizes and device types. Breakpoints are set to define when the content starts to adjust or change layout based on the device’s width.

Media queries can be used to check many things:

* width and height of the viewport
* width and height of the device
* Orientation
* Resolution

**Syntax**

@media not | only mediatype and (expression) {  
 // Code content  
}

**Example: In this example, we will use screen size in a media query to make the webpage responsive according to different breakpoints.**

<!DOCTYPE html>

<**html**>

<**head**>

<**title**>GeeksforGeeks</**title**>

<**style**>

.**gfg** {

**font-size**: 100px;

**font-weight**: **bold**;

**color**: **green**;

**text-align**: **center**;

}

.**geeks** {

**font-size**: 50px;

**text-align**: **center**;

}

**p** {

**font-size**: 25px;

**text-align**: **justify**;

}

*/\* styling for screen width less than 800 \*/*

@**media** **screen** **and** (**max-width**: **800px**) {

**body** {

**background-color**: **aqua**;

}

.**gfg** {

**font-size**: 50px;

}

.**geeks** {

**font-size**: 25px;

}

**p** {

**font-size**: 12px;

}

}

</**style**>

</**head**>

<**body**>

<**div** class="gfg">GeeksforGeeks</**div**>

<**div** class="geeks">HTML Introduction</**div**>

<**p**>

HTML stands for HyperText Markup Language. It is

used to design web pages using a markup

language. HTML is a combination of Hypertext and

Markup language. Hypertext defines the link

between web pages. A markup language is used to

define the text document within the tag which

defines the structure of web pages. This

language is used to annotate (make notes for the

computer) text so that a machine can understand

it and manipulate text accordingly. Most markup

languages (e.g. HTML) are human-readable. The

language uses tags to define what manipulation

has to be done on the text.

</**p**>

</**body**>

</**html**>

**Output: Background color and font size transition for width less than 800px.**

Video Player

00:00

00:09

**Note:**Sometimes, this method doesn’t show the correct output on Google Chrome.

**5. Responsive Layouts**

The responsive layout module of CSS includes the following properties.

**5.1 Using flexbox property**

In this approach, we will use [CSS display property](https://www.geeksforgeeks.org/css-display-property/) to make the page responsive. Display layouts like flexbox, inline, blocks, and grids can be used to make the design responsive. CSS flexbox property auto adjusts the content (no. of columns in a row) according to the screen width as shown in the output gif.

**Syntax**

.container{  
 display: flexbox;  
}

**Example: In this example, we will use display type flexbox to show adjust items automatically according to the screen size.**

<!DOCTYPE html>

<**html**>

<**head**>

<**title**>GeeksforGeeks</**title**>

<**style**>

**body** {

**background-color**: **aqua**;

}

.**gfg** {

**font-size**: 5vw;

**font-weight**: **bold**;

**color**: **green**;

**text-align**: **center**;

}

**button** {

**width**: 300px;

**font-size**: **larger**;

}

.**container** {

**display**: flexbox;

}

</**style**>

</**head**>

<**body**>

<**div** class="gfg">GeeksforGeeks</**div**>

<**div** class="container">

<**button**>HTML</**button**>

<**button**>CSS</**button**>

<**button**>JavaScript</**button**>

</**div**>

</**body**>

</**html**>

**Output**

Video Player

00:00

00:08

**Note:**Sometimes,this method doesn’t show the correct output on Google Chrome.

**5.2 Using CSS Grids**

This approach uses a [CSS display grid](https://www.geeksforgeeks.org/css-grid-layout-module/) to create a 2D layout along with other grid options. It allows us to decide the number of columns we want to keep and instead of rearranging the columns like Flexbox, it adjusts the content within individual column elements.

**Syntax**

.container{  
 display: grid;  
 /\* To define colums\*/  
 grid-template-columns: 1fr 1fr;   
}

**Example: In this example, CSS Grid layout is used to arrange contents in 2D form, i.e., rows and columns.**

<!DOCTYPE html>

<**html**>

<**head**>

<**title**>GeeksforGeeks</**title**>

<**style**>

**body** {

**background-color**: **aqua**;

}

.**gfg** {

**font-size**: 5vw;

**font-weight**: **bold**;

**color**: **green**;

**text-align**: **center**;

}

.**container** {

**font-size**: **x-large**;

**text-align**: **center**;

**display**: **grid**;

**grid-template-columns**: 1fr 1fr;

}

.**grid-item** {

**background-color**: rgb(220, 208, 232);

**border**: 2px **solid** rgb(70, 54, 84);

}

</**style**>

</**head**>

<**body**>

<**div** class="gfg">GeeksforGeeks</**div**>

<**div** class="container">

<**div** class="grid-item">HTML</**div**>

<**div** class="grid-item">CSS</**div**>

<**div** class="grid-item">JavaScript</**div**>

<**div** class="grid-item">Bootstrap</**div**>

</**div**>

</**body**>

</**html**>

**Output**

Video Player

00:00

00:07

**Note:**Sometimes,this method doesn’t show the correct output on Google Chrome.

**5.3 Using CSS MultiColumn**

It is similar to grids. CSS MulotiColumn allows developers to choose the properties like no. of columns, width, gap, etc. for each column. These values remain unchanged but the content inside the columns adjusts.

**Syntax**

.container{  
 column-count: 3; /\* Number of columns\*/  
 column-gap: 20px; /\* Gap between columns\*/  
 column-width: 200px; /\* Width of each column\*/  
 /\* Other column properties\*/  
}

**Example: This example uses CSS MultiColumn for content division into a specific number of columns.**

<!DOCTYPE html>

<**html**>

<**head**>

<**title**>GeeksforGeeks</**title**>

<**style**>

**body** {

**background-color**: **aqua**;

}

.**gfg** {

**font-size**: 5vw;

**font-weight**: **bold**;

**color**: **green**;

**text-align**: **center**;

}

.**container** {

**font-size**: **x-large**;

**text-align**: **left**;

**column-count**: 3;

**column-gap**: 5%;

}

</**style**>

</**head**>

<**body**>

<**div** class="gfg">GeeksforGeeks</**div**>

<**div** class="container">

<**div**>

HTML stands for HyperText Markup Language.

It is used to design web pages using a

markup language. HTML is a combination of

Hypertext and Markup language. Hypertext

defines the link between web pages. A markup

language is used to define the text document

within the tag which defines the structure

of web pages. This language is used to

annotate (make notes for the computer) text

so that a machine can understand it and

manipulate text accordingly. Most markup

languages (e.g. HTML) are human-readable.

The language uses tags to define what

manipulation has to be done on the text.

</**div**>

</**div**>

</**body**>

</**html**>

**Output**

Video Player

00:00

00:02

HTML Responsive Web Design allows websites to adapt to various screen sizes, ensuring a seamless user experience across devices. By using HTML and responsive techniques, designers and developers can create websites that look great whether viewed on a desktop or a smartphone.

This flexibility enhances accessibility and usability, making browsing a breeze for all users. In essence, HTML Responsive Web Design is important for ensuring that websites remain functional and visually appealing regardless of the device being used.

**HTML Responsive Web Design – FAQs**

**Why is Responsive Web Design important?**

*Responsive design ensures a consistent and optimal viewing experience across different devices, improving accessibility, user experience, and engagement.*

**How does a viewport meta tag help in Responsive Web Design?**

*It sets the width of web pages to match the device’s screen width, ensuring content adapts to different screen sizes.*

**What are some common layout techniques for responsive design?**

*Common techniques include using CSS Flexbox, CSS Grid, and MultiColumn layouts to adjust the content dynamically based on the screen size.*

**Can Responsive Web Design improve SEO?**

*Yes, it can improve SEO by providing a better user experience, reducing bounce rates, and ensuring that the site is accessible and usable across various devices, which search engines favor.*