

Media Queries:-

Viewport - the area of window or device.

media queries are used to set different style rules for different device or sized screens.

we use breakpoint to set the condition.

@ media (feature : value)

/* medium screen */

@ media (min-width : 600px) and

(max-width : 900px) {

 .container {

 background-color : lightgreen

 }

}

~~Animan~~

CLASS-2

* Math function *

Date: __/__/__

10 calc() function : this function performs as calculation. to be used property value.

div {

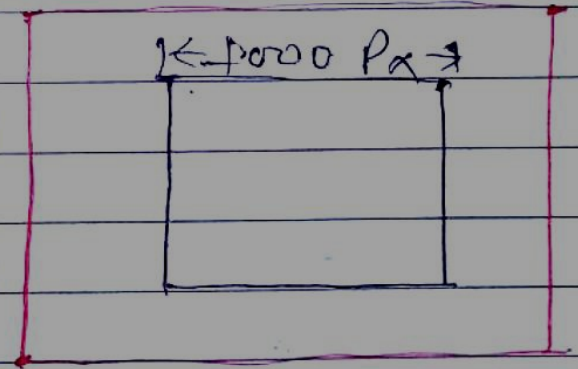
width : calc(100% - 100px.)

}

note - 100% mean total width of your parent element.

width : calc(100% - 200px)

↓ ↓
= (1200px - 200px)



width = 1000px ; Container : 1200px.

~~height~~ height : min(50%, 400px.) / max(50%, 400px.)

width : min(50%, 300px) ;

height : max(40%, 400px) ;

variables

Date: __/__/__

var() function is used to insert the value of a CSS variable:

Variable Creation & Utilise.

→ Creation:-

```
{  
  -- varName: (value);  
}
```

⇒ utilise `var(--varName);`

Transition in CSS.

CSS transitions allows you to change property values smoothly, over a given duration.

Ex. `div {`
 `width: 100px;`
 `background-color: red;`
 `transition: width 2s;`
 `}`

`div: hover {`

`width: 300px;`
`}`

#learnthessmarterway

* Speed Curve of the Transition

(i) transition-timing-function property specifies the speed curve of the transition effect.

- ease
- linear
- ease-in
- ease-out
- ease-in-out
- cubic-bezier(n, n, n, n)

```
#div { transition-timing-function: linear; }
```

2. Delay Transition :-

```
div {
```

```
    transition-delay: 1s;
```

```
}
```

3. Transition + Transformation

```
div {
```

```
    transition: width 2s, height 2s, transfo
```

```
}
```


More transition Ex. :-

```
div {  
  transition-property: width;  
  transition-duration: 2s;  
  transition-timing-function: linear;  
  transition-delay: 1s;  
}
```

Short hand property. =

```
div {  
  transition: width 2s linear 1s;  
}
```


CSS Animations

Date: __/__/__

Animation - gradually change from one style to another.

* Properties :-

- keyframes :
- animation-name :
- animation-duration :
- animation-delay :
- animation-iteration-count :
- animation-direction :
- animation-timing-function :
- animation-fill-mode :
- animation- :

⇒ @Keyframes Rule :-

The animation will gradually change from the current style to new style at certain times.

/x The animation code */

```
@keyframes box {  
  from { background-color: red; }  
  to { background-color: yellow; }  
}
```

#learnthesmarterway

/* The animation apply on element */

div {

width : 100px ;

height : 100px ;

background-color : red ;

animation-name : box ;

animation-duration : 4s ;

}

⇒ animation-duration : 3s ;

@keyframe box {

0% { background-color : red ; }

25% { background-color : blue ; }

100% { background-color : green ; }

}

The following example will change both the background-color and position of element.

@keyframe example {

0% {background-color: red; left: 0px; top: 0px;}

50% {background-color: yellow; left: 200px; top: 200px;}

etc...

}

div {

width: 100px;

position: relative;

animation-name: example;

animation-duration: 4s;

}

⇒ animation-delay: 2s;

delay for the start animation:

⇒ animation-iteration-count: 3;

number of times an animation should run.


```
div {
  ⇒ animation-direction : reverse ;
}
```

The animation-property can have following value.

- normal
- reverse
- alternate - Played forward 1st, then backward
- alternate-reverse - ~~forward~~ backward 1st, then forward.

```
div {
  ⇒ animation-timing-function : ease / linear / ease-in
    / ease-out / ease-in-out ;
}
```

It works same like ~~transition~~-timing-function

```
div {
  ⇒ animation-fill-mode : none / forward / backward
    / both ;
}
```

this property use for target element.

Date: __/__/__

⇒ Animation Shorthand -

div {

animation-name: example;

animation-duration: 5s;

animation-timing-function: linear;

animation-delay: 2s;

animation-iteration-count: 3;

animation-direction: alternate;

}

⇒ Shorthand -

div { animation: example 5s linear 2s infinite
alternate;

}