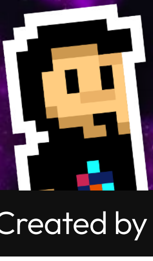


LENGUAJECSS.COM

CSS SYNTAX, COLORS AND UNITS

Created by @Manz (https://manz.dev/) https://lenguajecss.com/

CSS

C CSS Syntax

tag #id .class [attr] :pseudoclass ::pseudoelement { selector : value ; }

COMMENTS (ANNOTATIONS)

/* comment */

/* CSS is awesome */

S CSS Selectors

SELECTORS & COMBINATORS

PRIMARY

- # element with unique id
- . elements with class
- * all elements (universal)
- inside elements (descendant)

SECONDARY

- > direct childs
- + adjacents siblings
- ~ siblings (same level)

LOGICAL COMBINATORS

:is(A , B) group selectors

:where(A , B) less specific than :is()

:not(S) not match by S

:has(S) element w/ childs match by S

ATTRIBUTES

IGNORE CASE SENSITIVE [attr="..." i]

BASIC

- [attr] attr present
- [attr="val"] value is val
- [attr^="val"] begins w/ val
- [attr$="val"] ends w/ val

ADVANCED

- :not([attr]) attr absent
- [attr~="val"] contains val
- [attr|="es"] contains es-*
- [attr*="val"] includes val

PSEUDOCLASSES

LINKS

- :link :visited not/visited link
- :any-link links (visited or not)
- :target URL anchor target

LANG

- :lang(es) language match
- :dir(val) direction match

ACTION

- :hover user move mouse/pointer over element
- :active element is being activated by user
- :focus :focus-within :focus-visible w/ focus or not

USER FORM INTERFACE

- :enabled :disabled on/off input state
- :checked elements toggled "on" by user
- :indeterminate input on indeterminate state
- :read-only :read-write input modificable or not
- :placeholder-shown inputs w/ active placeholder
- :default default elements for form (input, option...)
- :valid :invalid userdata pass check validation
- :user-valid :user-invalid idem, with user interaction
- :in-range :out-of-range userdata ok/out of range
- :required :optional required/optional for submit

STRUCTURAL

CHILDS

- :first-child
- :last-child
- :nth-child(n)
- :nth-last-child(n)
- :only-child

CHILDS (SAME)

- :first-of-type
- :last-of-type
- :nth-of-type(n)
- :nth-last-of-type(n)
- :only-of-type

n odd even n of S

OTHERS

- :root root element of the document
- :host root element of shadow dom
- :empty element without contents

PSEUDOELEMENTS

CONTENT

- ::before / ::after
- ::first-line
- ::first-letter t ext
- ::file-selector-button
- ::marker list sign

HIGHLIGHT

- ::selection selected text of user
- ::target-text fragment url style
- ::spelling-error ::grammar-error
- ::backdrop background elements
- ::placeholder form text hints

C CSS Colors

COLOR PROPERTIES

color: CanvasText color

opacity: 1 alpha

COLOR MODELS

alpha 0...1 0%...100%

KEYWORDS

keyword transparent currentColor

COLOR FORMATS

RGB SPACE

rgb(25% 41% 88% / 50%) ~~rgba()~~	RGB / A
rgb(65 106 225 / 50%)	RGB / A
# 41 6A E1 88	#RRGGBBAA / #RGBA
hsl(120deg 25% 75% / 50%) ~~hsla()~~	HSL / A
hwb(120deg 55% 25% / 50%)	HWB / A

DEVICE SPACE

lab(41% 60 42 / 50%)	BW RG BY / alpha
oklab(51% 0.2 0.1 / 50%)	BW RG BY / alpha
lch(41% 99 35deg / 50%)	BW S T / alpha
oklch(50% 0.2 26deg / 50%)	BW S T / alpha

COLOR FUNCTIONS / OPERATIONS

SPACE SRGB SRGB-LINEAR DISPLAY-P3 A98-RGB PROPHOTO-RGB REC2020 XYZ XYZ-D65 XYZ-D50

color(space p1 p2 p3) color in a spacecolor

light-dark(color lightmode , color darkmode)

RELATIVE COLORS

color-mix(in space , c1 % , c2 %) mix colors

rgb(from color , colors / A) color-based

U CSS Values & Units

RESETTING ALL PROPERTIES

all: initial value inherit parent unset previous

UNITS

ABS

- px pixel cm mm Q ¼mm in inches pc pt

REL

- % parent size em rem root em ex rex cap
- rcap ch 1 char rch ic ric lh 1 line rlh

VIEWPORT

- vw vh vmin vmax vi vb % viewport
- svw svh svmin svmax svi svb small
- lvw lvh lvmin lvmax lvi lvb large
- dvw dvh dvmin dvmax dvi dvb dynamic

CQ

- cqw cqh cqmin cqmax cqi cqb container

V CSS Variables

CUSTOM PROPERTIES

-- varname : value ; declaring vars

var(-- varname , fallback ...) using vars

env(environment-variable) user-agent vars

LENGUAJECSS.COM

BOX MODEL AND BORDERS

Created by @Manz (https://manz.dev/) https://lenguajecss.com/

CSS

M CSS Box Model

INTRINSIC SIZES

max-content min-content fit-content

DIMENSIONS

width: auto size %
min-width: 0 size %
max-width: none size %

height: auto size %
min-height: 0 size %
max-height: none size %

MARGIN/PADDINGS

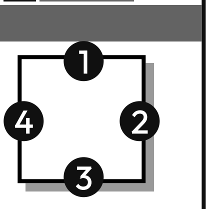
SPECIFIC MARGINS

margin-top: 0 size %
margin-right: 0 size %
margin-bottom: 0 size %
margin-left: 0 size %

padding-top: 0 size %
padding-right: 0 size %
padding-bottom: 0 size %
padding-left: 0 size %

MARGIN/PADDING SHORTHAND

margin/padding: [❶] [❷] [❸] [❹]
margin/padding: [❶] [❷❹] [❸]
margin/padding: [❶❸] [❷❹]
margin/padding: [❶❷❸❹]



OVERFLOW/VISIBILITY

overflow-x: visible hidden scroll auto
overflow-y: visible hidden scroll auto
overflow: overflow-x overflow-y

VISIBILITY

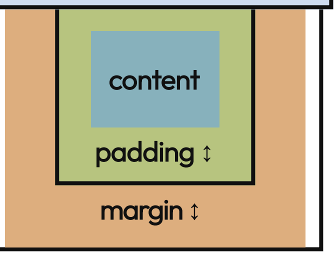
visibility: visible hidden collapse

MODEL BOX TYPES

display: inline none
inline-block block
inline-list-item list-item
inline-table table
table-cell table-row
inline-grid grid subgrid
inline-flex flex

CHANGE BOX MODEL

box-sizing: content-box border-box



BI CSS Border Image

BORDER IMAGE

border-image-source: none url(img.ext)
border-image-slice: 100% top right bottom left fill
border-image-width: 1 top right bottom left
border-image-outset: 0 top right bottom left
border-image-repeat: stretch repeat round space
border-image: source slice /width outset repeat

BORDER IMAGE VALUES

-slice → number %
-width → auto size number %
-outset → number size
-repeat → repeat space round no-repeat

T CSS Tables

TABLES PROPERTIES

border-collapse: separate collapse
border-spacing: 0 size
caption-side: top bottom
empty-cells: show hide
table-layout: auto fixed

B CSS Borders

BORDERS

border-width: size thin medium thick
border-style: none border-style hidden
border-color: currentColor color

SHORTHANDS

border-top-*
border-right-*
border-bottom-*
border-left-*

SIDE BORDER SHORTHAND

border-top: width style color
border-right: width style color
border-bottom: width style color
border-left: width style color

MAIN SHORTHAND

border: width style color

BORDER STYLES

solid	dotted
dashed	double
groove	ridge
inset	outset

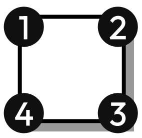
C Rounded Corners

SPECIFIC CORNERS

border-top-left-radius: 0 size %
border-top-right-radius: 0 size %
border-bottom-left-radius: 0 size %
border-bottom-right-radius: 0 size %

MAIN SHORTHAND

border-radius: [❶][❷][❸][❹]
border-radius: [❶][❷❹][❸]
border-radius: [❶❸][❷❹]
border-radius: [❶❷❸❹]



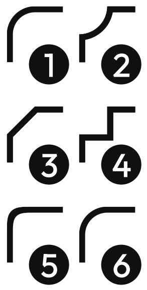
MAIN SHORTHAND WITH HORIZONTAL/VERTICAL RADIUS

border-radius: border-radius / border-radius

CS Corner Shape

CORNER SHAPE VALUES

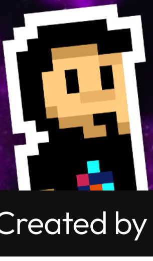
corner-shape:
❶ round ~ superellipse(2)
❷ scoop ~ superellipse(0.5)
❸ bevel ~ superellipse(1)
❹ notch ~ superellipse(0)
❺ squircle ~ superellipse(4)
❻ straight ~ superellipse(Infinity)
superellipse(n) ~ custom shape corner



L CSS Lists

LISTS

list-style-image: none url(img.png)
list-style-position: inside outside
list-style-type: disc circle square upper-alpha
lower-alpha upper-roman lower-roman decimal
decimal-leading-zero lower-greek armenian
georgian none countername symbols(...)
list-style: type position image



LENGUAJECSS.COM

FONTS, TYPOGRAPHY AND TEXTS

Created by @Manz (https://manz.dev/) https://lenguajecss.com/

CSS

F Base Fonts

FONTS (TYPOGRAPHY)

font-family: font1 , font2 , font3 ... , safe-font ;

font-size: size % smaller larger
xx-small x-small small medium
large x-large xx-large

font-size-adjust: none number

font-style: normal italic oblique

font-synthesis: none weight style

font-variant: none small-caps

font-weight: normal bold lighter bolder
number 1~1000

font-width: normal % condensed expanded
ultra-condensed extra-condensed semi-condensed
semi-expanded extra-expanded ultra-expanded

font: style variant weight width size/
line-height family caption icon menu
message-box small-caption status-bar

FF Font Face

FONT LOADING

```
@font-face {
  font-family: font-name;
  font-display: swap;
  src: url(file.woff2) format("woff2"),
   url(file.woff) format("woff"),
   url(file.ttf) format("truetype");
  unicode-range: U+000-27FF;
}
```

CDN FONTS

https://fonts.googleapis.com/css2?family=Open+Sans:wght@100..800&family=...
https://cdn.jsdelivr.net/fontsource/fonts/roboto:vf@latest/latin-wght-normal.woff2

CT CSS Text

TRANSFORMING TEXT

text-transform: none capitalize uppercase lowercase full-width

WHITE SPACES & BREAKING WORDS

white-space-collapse: break-spaces collapse preserve preserve-breaks

text-wrap-mode: nowrap wrap

white-space-trim: none discard-before discard-after discard-inner

white-space: normal collapse wrap-mode trim

tab-size: 8 number size

LINE BREAKING & WORD BOUNDARIES

line-break: auto loose normal strict anywhere

word-break: normal keep-all break-all break-word

hyphens: none manual auto

overflow-wrap: normal break-word anywhere

TEXT-WRAP

text-wrap-style:
auto balance
pretty stable

ALIGNMENT & JUSTIFICATION

text-align: left right center justify start end match-parent

text-align-last: auto left right center justify start end match-parent

text-justify: auto none inter-word inter-character

SPACING

word-spacing: normal size letter-spacing: normal size

EDGE EFFECTS

text-indent: 0 size hanging each-line

hanging-punctuation: none first last force-end allow-end

SHADOW EFFECTS

text-shadow: none pos-x pos-y blur color

box-shadow: none pos-x pos-y blur spread color inset

TD CSS Text Decoration

LINE OPTIONS

underline overline line-though

LEVEL 2 (LEGACY)

text-decoration: none line-option

LEVEL 3

text-decoration-line: none line-option blink

text-decoration-style: solid double dotted dashed wavy

text-decoration-color: currentColor color

text-decoration-skip: none objects spaces ink edges
box-decoration

text-decoration: none line style color

text-underline-position: auto under left right

EMPHASIS MARKS

text-emphasis-style: none string status [dot circle
double-circle triangle sesame]

text-emphasis-color: currentColor color

text-emphasis-position: over under left right

text-emphasis: style color

STATUS

filled open

EX: open circle;

MC CSS Multi Column

COLUMNS

column-width: auto size

column-count: auto number

columns: width counter

SPANNING & FILLING

column-span: none all

column-fill: auto balance

COLUMN RULES

column-rule-width: size thin medium thick

column-rule-style: style none

column-rule-color: color

column-rule: width style color

COLUMN BREAKS

break-before/break-after: auto left right
always recto verso avoid page column
avoid-page avoid-column

break-inside: auto avoid avoid-page
avoid-column

orphans: 2 number widows: 2 number

LENGUAJECSS.COM

BACKGROUNDS, IMAGES AND RULES

Created by @Manz (https://manz.dev/) https://lenguajecss.com/

CSS

CB CSS Backgrounds

BACKGROUND BASE PROPERTIES

background-color: transparent currentColor color

background-image: none url(im1.png), url(im2.png)...

background-repeat: also support 2 parameters

repeat repeat-x repeat-y space round no-repeat

background-attachment: scroll fixed local

BACKGROUND POSITION

background-position-x: 0% size %

left center right x-start x-end

background-position-y: 0% size %

top center bottom y-start y-end

background-position: 0% 0% pos-x pos-y

COLOR-STOP

color Basic color

color start end

BACKGROUND MODERN PROPERTIES

background-clip:

border-box padding-box content-box text

background-origin: border-box padding-box content-box

background-size: auto width height contain cover

background: color position size repeat origin clip attachment image

RADIUS SIZE

closest-side

farthest-side

closest-corner

farthest-corner

CR CSS Rules

IMPORTING SYNTAX AND ALTERNATIVE SYNTAX

@import "file.css" @import url("file.css")

IMPORTING FEATURES — ONLY ON TOP FILE

@import "file.css" print — apply styles if printing

@import "file.css" MQ conditions — if media query is true

@import "file.css" supports(condition) — if feature supported

@import "file.css" layer(layername) — apply styles on layer

CASCADE LAYERS — CAN NEST LAYERS

@layer { ... } create anonymous cascade layer

@layer layername1, layername2, ... ; — order layers

@layer layername1, layername2, ... { ... } — create/order layers

@layer layername.sub-layername { ... } — create sublayer

SCOPING STYLES

@scope { ... } limit scope from parent to child (using on inline <style>)

@scope (selector) { ... } — limit scope from selector to child

@scope (selector) to (selector) { ... } — limit between selectors

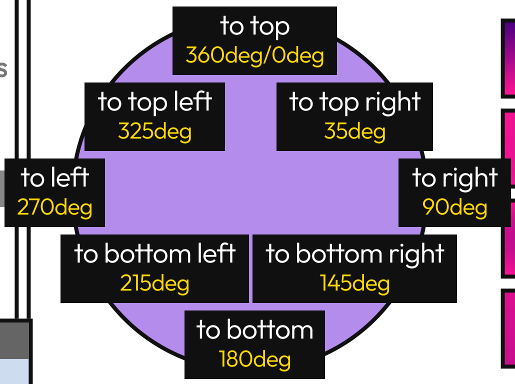
NESTING STYLES

```
A-selector { parent selector
  B-selector { ... } equivalent to "A B"
  & B-selector { ... } equivalent to "A B"
  B-selector & { ... } equivalent to "B A"
  @media ( condition ) { ... } nesting media query
  @scope ( selector ) { ... } nesting scoping
}
```

CG CSS Gradients

GRADIENTS

DIRECTIONS



LINEAR GRADIENTS

background-image:

linear-gradient(direction, color-stop1, ...)

repeating-linear-gradient(...same params...)

RADIAL GRADIENT (CIRCLE)

background-image:

radial-gradient(circle rsize at pos, col, ...)

repeating-radial-gradient(...same params...)

RADIAL GRADIENT (ELLIPSE)

background-image:

radial-gradient(ellipse rx ry at pos, col, ...)

repeating-radial-gradient(...same params...)

CONIC GRADIENTS

background-image:

conic-gradient(from angle at pos, col1, ...)

repeating-conic-gradient(...same params...)

CI CSS Images

FUNCTIONS

image-set(— set alternative image

url(im1.avif) type("image/avif"), — by type

url(im2.webp) 2x,... — by density

);

cross-fade([image | color] size %, ...)

IMAGES AND OBJECTS

SIZING

object-fit: fill contain cover none scale-down

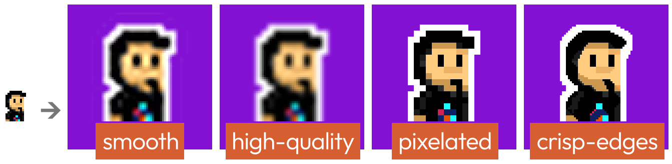
object-position: 50% 50% background-position

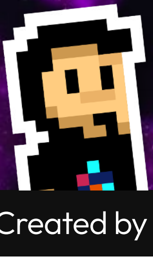
object-view-box: rect() inset() xywh()

IMAGE PROCESSING

image-orientation: from-image none

image-rendering:





LENGUAJECSS.COM

EFFECTS, FILTERS AND SHAPES

Created by @Manz (https://manz.dev/) https://lenguajecss.com/

CSS

CF CSS Filters

FILTER PROPERTY

filter / backdrop-filter: none filter1 filter2 filter3 ... ;

FILTERS

blur(blur)

brightness(number %)

contrast(number %)

saturate(number %)

grayscale(number %)

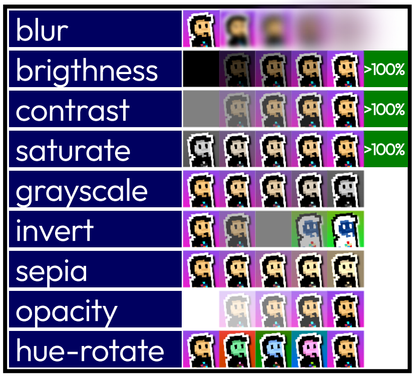
invert(number %)

sepia(number %)

opacity(number %)

hue-rotate(angle)

drop-shadow(pos-x pos-y blur color) (= text-shadow)



CM CSS Masks

CLIPPING SHAPE

clip-path: none url(img.png) basic-shape shape-box

POSITIONED MASKS

mask-image: none url(img.svg#mask) gradient

mask-mode: auto alpha luminance

mask-repeat: no-repeat background-repeat

mask-position: center background-position

mask-clip: border-box no-clip shape-box

mask-origin: border-box shape-box

mask-size: auto background-size

mask-composite: add subtract intersect exclude

mask: image mode position /size repeat clip composite

C Compositing

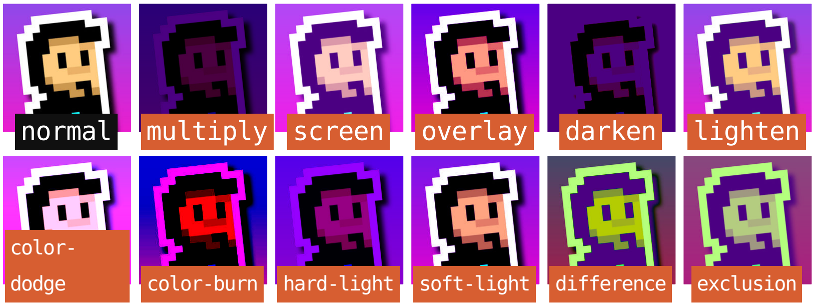
COMPOSITING AND BLENDING

background-blend-mode: normal blend

mix-blend-mode: normal blend isolation: auto isolate

BLEND MODES

SEPARABLE



NON SEPARABLE

hue
saturation
color
luminosity

CSO CSS Shape Outside

FLOATING AREA SHAPE

shape-outside: none url(img.png) gradient basic-shape shape-box

shape-image-threshold: 0.0 number

shape-margin: 0 size %

CS CSS Shapes

BASIC SHAPES

RECTANGLE

rect(x w y h round radius)

inset(top right bottom left round radius)

xywh(x y w h round radius)

CIRCLE/ELLIPSE

circle(shaperadius at position)

ellipse(radius-x radius-y at position)

ADVANCED

polygon(x0 y0 , ... , xi yi)

path(svg coords)

shape(from position , shape commands)

SHAPE COMMANDS

MOVE COMMANDS

from x y move to x y move by x y

close close shape and connect with first point

LINE COMMANDS

line to x y hline to x vline to y

CURVE COMMANDS

curve to x y reuse previous point

curve to x y with Ax Ay use control point A

curve to x y with Ax Ay / Bx By ctrl point A,B

SYMETRIC CURVE COMMANDS

smooth to x y reuse previous point

smooth to x y with Ax Ay use control point A

SEGMENT COMMANDS

arc to x y implicit arc (radius and shape autocalc)

arc to x y of r between current point and x,y

arc to x y of Rx Ry elliptic arc with radius R

OPTIONS

cw ccw clockwise or counterclockwise

large small segment size rotate angle (def: 0deg)

CP CSS Performance

PERFORMANCE PROPERTIES

will-change: property optimize painting/animation

contain: none strict content size inline-size layout style paint

content-visibility: visible auto hidden

LENGUAJECSS.COM

TRANSITIONS, ANIMATIONS AND TRANSFORMS

CSS

Created by @Manz (https://manz.dev/)   https://lenguajecss.com/

CT CSS Transitions

TRANSITIONS

transition-property: all none property1 , property2 , ...

transition-duration: 0s time

transition-timing-function: ease timing-function

transition-delay: 0s time

transition: property duration t-function delay

TIMING FUNCTIONS

TIMING BASE FUNCTIONS	
ease	(0.25, 0.1, 0.25, 1)
linear	(0.00, 0.0, 1.00, 1)
ease-in	(0.42, 0.0, 1.00, 1)
ease-out	(0.00, 0.0, 0.58, 1)
ease-in-out	(0.42, 0.0, 0.58, 1)
cubic-bezier()	custom timing func
linear()	custom: linear(0, 1) = linear

TIMING STEP FUNCTIONS	
step-start	steps(1,start)
step-end	steps(1,end)
steps(n)	steps(n,end)
steps(a,b)	steps(a,b)

CHAINED ANIMATIONS

name1 5s linear ,
name2 5s linear 5s ,
name3 6s linear 10s ;

CA CSS Animations

ANIMATIONS

animation-name: none name1 , name2 , ...

animation-duration: 0s time

animation-timing-function: ease timing-function

animation-delay: 0s time

animation-iteration-count: 1 number infinite

animation-direction: normal reverse alternate alternate-reverse

animation-fill-mode: none forwards backwards both

animation-play-state: running paused

animation-range-start: normal name size|% size|%

animation-range-end: normal name size|% size|%

animation-range: range-start range-end

animation-composition: replace add accumulate

animation-timeline: auto none --name scroll() view()

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

SCROLL DRIVEN ANIMATION

TIMELINE FUNCTIONS

scroll() → scroll([nearest root self] [x y])

view() → view([x y] auto size-start size-end)

TIMELINE PROPERTIES

SCROLL

scroll-timeline-name: none --name

scroll-timeline-axis: x y

scroll-timeline: name axis

VIEW

view-timeline-name: none --name

view-timeline-axis: x y

view-timeline-inset: auto size|% size|%

view-timeline: name axis inset

KEYFRAMES

```
@keyframes name {
  0% { /* css */ }
  100% { /* css */ }
}
```

CT CSS Transforms

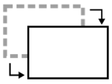
TRANSFORM PROPERTY

transform: none func1 func2 ...

2D TRANSFORM FUNCTIONS

translateX(size|%) move x-axis

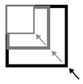
translateY(size|%) move y-axis



translate(size|% , size|%)

scaleX(number) resize x-axis

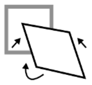
scaleY(number) resize y-axis



scale(number , number)

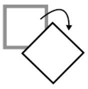
skewX(angle) horizontal shear transform

skewY(angle) vertical shear transform



skew(angle , angle)

rotate(angle) spin element angle



matrix(n1 , n2 , n3 , n4 , n5 , n6)

3D TRANSFORM FUNCTIONS

translateZ(size) move along depth

translate3d(size|% , size|% , size)

scaleZ(number) stretch along depth

scale3d(number , number , number)

rotateX(angle) spin on x-axis

rotateY(angle) spin on y-axis

rotateZ(angle) spin on z-axis

rotate3d(number , number , number , angle)

TRANSFORM OPTIONS

transform-origin: 50% 50% pos-x pos-y pos-z

transform-style: flat preserve-3d preserve 3d child

backface-visibility: visible hidden hide rear face

PERSPECTIVE OPTIONS

perspective: none size depth illusion control

perspective-origin: 50% 50% pos-x pos-y pos-z

INDIVIDUAL PROPERTIES

TRANSLATE PROPERTY

translate: size|% move only x axis

translate: size|% size|% only x/y axis

translate: size|% size|% size x/y/z axis

SCALE PROPERTY

scale: num|% apply same factor to x/y axis

scale: num|% num|% apply to x/y axis

scale: num|% num|% num|% x/y/z axis

ROTATE PROPERTY

rotate: angle apply to z axis

rotate: axis angle ex: rotate: y 10deg

rotate: x y z angle ex: rotate: 1 1 0.5 45deg

LENGUAJECSS.COM

LAYOUTS, FLEX & GRID

Created by @Manz (https://manz.dev/)

https://lenguajecss.com/

CSS

F CSS Flex

PARENT PROPERTIES

ORDERING AND ORIENTATION

flex-direction: row column row-reverse column-reverse

flex-wrap: nowrap wrap wrap-reverse

flex-flow: direction wrap

PRIMARY AXIS

justify-content: start center end space-between space-around space-evenly

SECONDARY AXIS

align-items: start center end baseline stretch

MULTI-LINE CONTAINER (WRAP)

align-content: start center end stretch space-between space-around space-evenly

CHILD PROPERTIES

FLEXIBILITY

flex-grow: 0 number

flex-shrink: 0 number

flex-basis: auto content size %

flex: grow shrink basis

GRID AREA VALUES

area	area name
.	empty area
none	no defined

ALIGNMENT AND ORDERING

align-self: auto start center end baseline stretch

order: 0 number — reorder items with weight

P CSS Position

POSITIONING

position: static relative absolute fixed sticky

FLOATING

float: none left right

clear: none left right both

POSITIONING (NO STATIC)

top: auto size %

left: auto size %

right: auto size %

bottom: auto size %

z-index: auto number

G CSS Gaps

GUTTERS (GRID AND FLEX GAPS)

row-gap: 0 size %

column-gap: 0 size %

gap: row-gap column-gap

AREA PREFIXES

span-all / span-*

x-* / y-*

span-x-*

span-y-*

x-self-* / y-self-*

span-x-self-*

span-y-self-*

AP CSS Anchor Position

ANCHOR SETTING

anchor-name: none --name

anchor-scope: none all --name

position-anchor: auto --name

ANCHOR POSITIONING

position-area: top left right bottom

anchor(--name top left right bottom)

ANCHOR SIZING

anchor-size(width height block inline)

G CSS Grid

PARENT PROPERTIES

EXPLICIT GRID

grid-template-columns: none column1 column2 ...

grid-template-rows: none row1 row2 ...

grid-template: rows / columns

GRID VALUES (ROWS AND COLUMNS)

auto size % fr	automatic size or size/fraction unit
repeat(number , size)	repeat fragments n times
repeat(auto-fill , size)	repeat + fill space & empty
repeat(auto-fit , size)	repeat + fit + remove empty
minmax(min , max)	min <= size range <= max

GRID AREA

grid-template-areas: " area1 area2 " ...

GRID ALIGMENT

justify-content: normal start center end stretch space-between space-around space-evenly

align-content: normal start center end stretch space-between space-around space-evenly

place-content: align-content justify-content

justify-items: normal start center end stretch

align-items: normal start center end stretch baseline

place-items: align-items justify-items

IMPLICIT GRID

grid-auto-columns: auto size % — auto-created columns size

grid-auto-rows: auto size % — auto-created rows size

grid-auto-flow: row column row dense column dense

CHILD PROPERTIES

CELLS PLACEMENT

grid-column-start: auto line — column start line

grid-column-end: auto line — column end line

grid-column: column-start / column-end

grid-row-start: auto line — row start line

grid-row-end: auto line — row end line

grid-row: row-start / row-end

LINE

number [linename]	refer to a numbered or named grid line
span number [linename]	span across until grid line

CHILD ALIGMENT

justify-self: auto normal stretch start center end

align-self: auto normal stretch start center end

place-self: align-self justify-self

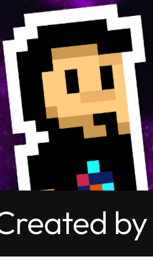
AREA PLACEMENT

grid-area: area-name named area

order: 0 number

GLOBAL SHORTHAND

grid: row-start / column-start / row-end / column-end



LENGUAJECSS.COM

MATHEMATICS AND CSS LOGIC

CSS

Created by @Manz (https://manz.dev/)   https://lenguajecss.com/

<meta name="viewport" content="initial-scale=1, width=device-width">

M CSS Math

CÁLCULOS

calc(operations)	calc operations

`calc(var(--value) + 50px) , calc(25px + 50%)`

SIGN RELATED

abs(number)	get absolute value
sign(number)	sign of number (-1, 0 or 1)

COMPARE FUNCTIONS

min(A , B , ...)	max(A , B , ...)
clamp(A , B , C)	~ max(A, min(B, C))

STEPPED FUNCTIONS

round(method , value , interval)	
method → nearest up down to-zero	
mod(A , B)	modulus operation
rem(A , B)	remainder of trunc division

TRIGONOMETRIC FUNCTIONS

sin(angle) asin(number)

cos(angle) acos(number)

tan(angle) atan(number) atan2(Y , X)

EXPONENTIAL FUNCTIONS

pow(A , B) sqrt(A) hypot(A , ...)

log(A , B) exp(A)

F CSS Functions

FUNCTION DEFINITION SYNTAX

```
@function --func-name ( --p1 , --p2 , ... ) {
  result: value ; you can use params: var(--p1)
}
```

USE CSS FUNCTION

```
property: --func-name( p1 , p2 , ... );
```

R CSS Random

RANDOM VALUE FUNCTIONS

random(min , max)	cached random value
random(min , max , by step)	step-random
random(--v , min , max)	non-cached
random(per-element , mn , mx)	random

RANDOM ITEMS FUNCTIONS

random-item(v1 , v2 ,...)	cached random item
random-item(--v , v1 , v2 ,...)	non-cached
random-item(per-element , v1 , v2 ,...)	random

WE CSS When/Else

CONDITIONAL RULES

@when (condition) { ... }	conditional rule support
@else (condition) { ... }	conditional else support

C CSS Conditionals

MEDIA QUERIES

@media (condition) { ... }	standard media query
@media (cond1) and (cond2) { ... }	multi-condition MQ
@media not (condition) { ... }	negative media query
@media print { ... }	print media query

MEDIA FEATURES (CONDITIONS)

SCREEN FEATURES

width/height: size	device width/height size
aspect-ratio: number / number	size proportion
orientation: landscape portrait	device screen rotation
overflow-block: none scroll paged	primary axis scrollability
overflow-inline: none scroll	secondary axis scrollability
display-mode: fullscreen picture-in-picture browser minimal-ui standalone	PWA presentation mode
scripting: none initial-only enabled	scripting support level

PRECISION FEATURES

pointer: none coarse fine	detect primary device
any-pointer: none coarse fine	detect any device
hover: none hover	detect primary device
any-hover: none hover	detect any device

OTHERS FEATURES

resolution: ddpx-res infinite	pixel density
update: none slow fast	speed update screen
color: number	color-depth (8 bits, 16 bits, ...)

CONTAINER QUERIES

@container name (condition) { ... }	MQ for containers
@container name style(prop: value) { ... }	style check

CONTAINER QUERIES PROPERTIES

container-name: none name	container name
container-type: normal size inline-size	axis (block or inline)
container: container-name / container-type	

CONDITIONAL IF

if(true cond : true ; else: false)	css conditional
if(cond1 : true ; cond2 : true ; ...; else: false)	multi
if(style(prop: value) : true ; else: false)	style check
if(supports(prop: value) : true ; else: false)	support check
if(media(condition) : true ; else: false)	media style check

SUPPORTS CONDITIONALS

@supports (condition) { ... }	check «prop: values» support
@supports not (condition) { ... }	negative check support
@supports (cond1) and or (cond2) { ... }	multi check
@supports selector(selector) { ... }	check complex selector
@supports font-tech(feature) { ... }	(variations, palettes, ...)
@supports font-format(format) { ... }	woff, woff2, ...