

**ORACLE
TECH DAYS**

ORACLE



Welcome to Oracle Week 2024

we will start
in a few minutes



ScSS + CSS



Styling Modern Applications

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

Sass

ScSS

+

CSS



Hi, I'm...



Kobi Hari

- Freelancer
- Developer, Instructor and Consultant
- Angular, Async development, .NET core



My email:

hari@applicolors.com



Courses on Udemy: <https://www.udemy.com/user/kobi-hari/>



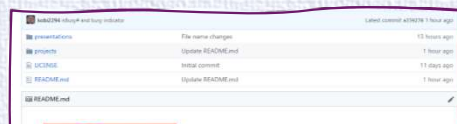
My Angular Channel: <https://www.youtube.com/@kobihari>

ORACLE

JOHN BRYCE
תלמדו היטב. זה עובד!
a matrix company

We have a GitHub Repository!

[kobi-hari-courses/2411-oracle-tech-days-sassy-apps](https://github.com/kobi-hari-courses/2411-oracle-tech-days-sassy-apps)



A screenshot of the GitHub repository's file browser. It shows a list of files and folders: 'presentations', 'projects', 'LICENSE', and 'README.md'. Each item has a file icon, its name, and a 'Latest commit' timestamp.

File name	Latest commit	Time ago
presentations	File name changes	13 hours ago
projects	Update README.md	1 hour ago
LICENSE	Initial commit	11 days ago
README.md	Update README.md	1 hour ago

Session Summary

Commits


Application Setup


#	Link	Description
1	Initial application	Created an empty application using the angular CLI: <code>ng new redux-pop-quizz</code>
2	Added Material Design	Added angular material using the cli: <code>ng add @angular/material</code>
3	Custom Theme	Defined the theme palettes in partial <code>_common.scss</code> and generated the material styles in <code>styles.scss</code>
4	Tested theme	Used some angular material components like icon and button to test the new theme
5	Grid, Flex layout	Created layout <i>between search and add button</i> in either grid or flex for easier layout management
6	CSS Variables	Added CSS variables


Project Code with Commits Summary


Building the UI

#	Link	Description
7	Toolbar	Added material toolbar

 [presentations](#)

 [projects](#)

 [LICENSE](#)

 [README.md](#)

This presentation and other Useful Links

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

Sass

ScSS

+

CSS



Our Agenda



CSS Tricks – Selectors, Flex and Grid



CSS Custom Properties



ScSS Style Preprocessor



Themeing Applications

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

Sass

ScSS

+

CSS



CSS Tricks

Selectors

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



ScSS

+

CSS



Selectors we all know

selectors.css

```
/* type selectors */
div {
  color: green;
}

/* class selector */
.nice {
  background-color: lime;
}

/* id selector */
#section-1 {
  text-decoration: underline;
}
```

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



ScSS

+

CSS



Selectors some of us know

selectors.css

```
/* universal */
* {
  font-weight: bold;
}

/* attribute selector */
[disabled] {
  color: gray;
}

/* attribute value selector */
[type="number"] {
  font-size: 16px;
}
```

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



ScSS

+

CSS



Attribute Selectors

attributes.css

```
/* Starts with */  
[href^='https:']
```

```
/* Ends with */  
[href$='.com']
```

```
/* Substring */  
[href*='-']
```

attributes.css

```
/* space seperated list */  
[tags~='styling']
```

```
/* dash seperated list */  
[file-name |='user']
```

```
/* case insensitive */  
[status="open" i]
```

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



ScSS

+

CSS



Combinators

combinators.css

```
/* Descendants */  
.section-1 img {}  
  
/* Direct Child */  
.section-1>img {}  
  
/* And */  
.large.expensive {}  
  
/* Or */  
.large, .expensive {}
```

combinators.css

```
/* following */  
p~img {}  
  
/* directly following */  
p+img {}
```

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



ScSS

+

CSS



Pseudo Class

pseudo-class.css

```
/* on hover */
: hover {}

/* on visited */
a: visited {}

/* Active */
a: active {}

/* When in focus */
input: focus {}

/* When disabled */
input: disabled {}
```

pseudo-class.css

```
/* Child counting */
: nth-child(3) {}

: nth-child(-2n + 5) {}

/* Child of type */
: nth-of-type(4) {}

/* Negation */
: not(.hot) {}
```

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



ScSS

+

CSS



Pseudo Elements

pseudo-element.css

```
/* First letter */
p::first-letter {}

/* First-line */
p::first-line {}

/* Selection */
p::selection {}

/* Bullet Marker */
li::marker {}
```

pseudo-element.css

```
/* First Child (before content) */
div::before {
  content: 'XYZ';
}

/* Last Child (after content) */
div::after {
  content: 'Thank you';
}
```

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

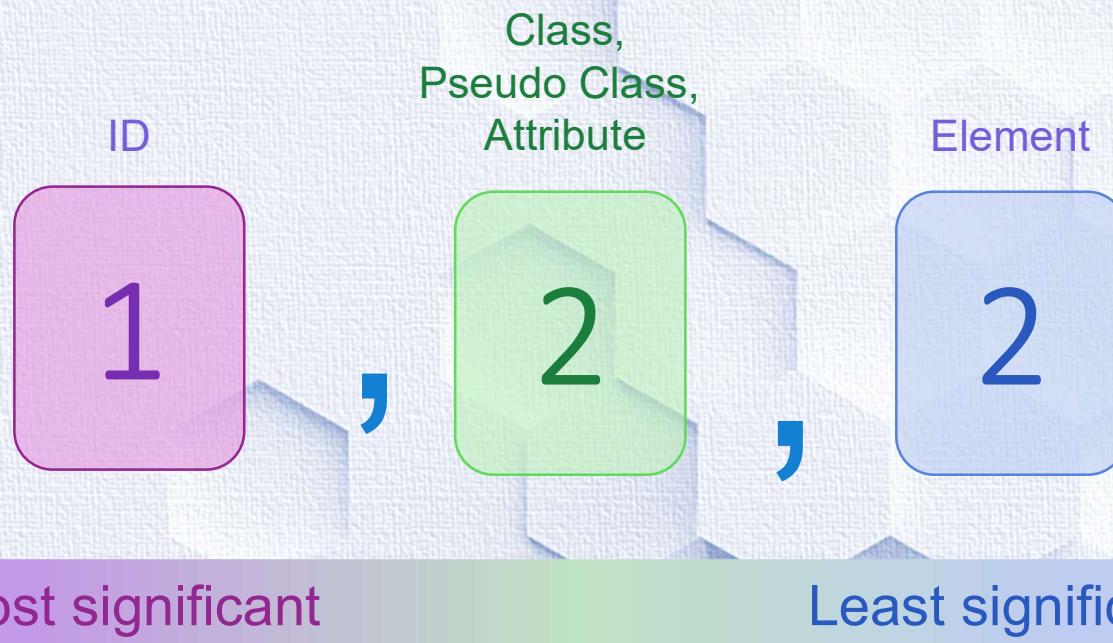
Specificity

Sass

ScSS

+

CSS



#first-section div.data input[disabled]

Sass

ScSS

+

CSS



CSS Tricks

Flex Boxes

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



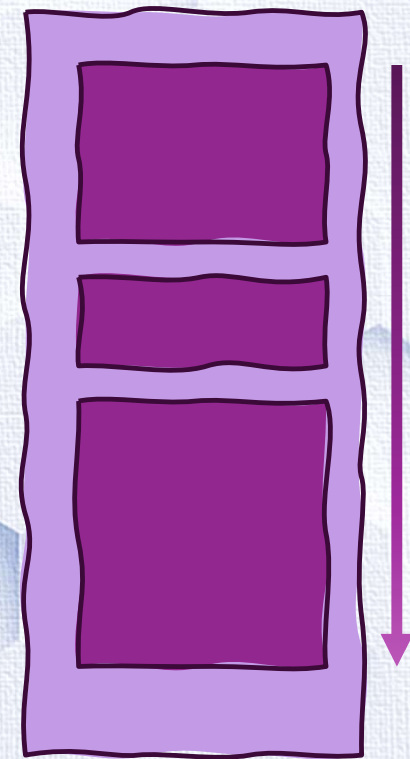
ScSS + CSS



What is flex?

- Flex is a **layout** that aligns and distributes **items** in a **container**.
- Enables flexible **spacing** and **resizing** of child elements.
- Supports **responsive** layouts with minimal CSS code.

display: flex



ORACLE

JOHN BRYCE
תלמדו היטב. זה עובד!
a matrix company



ScSS

+

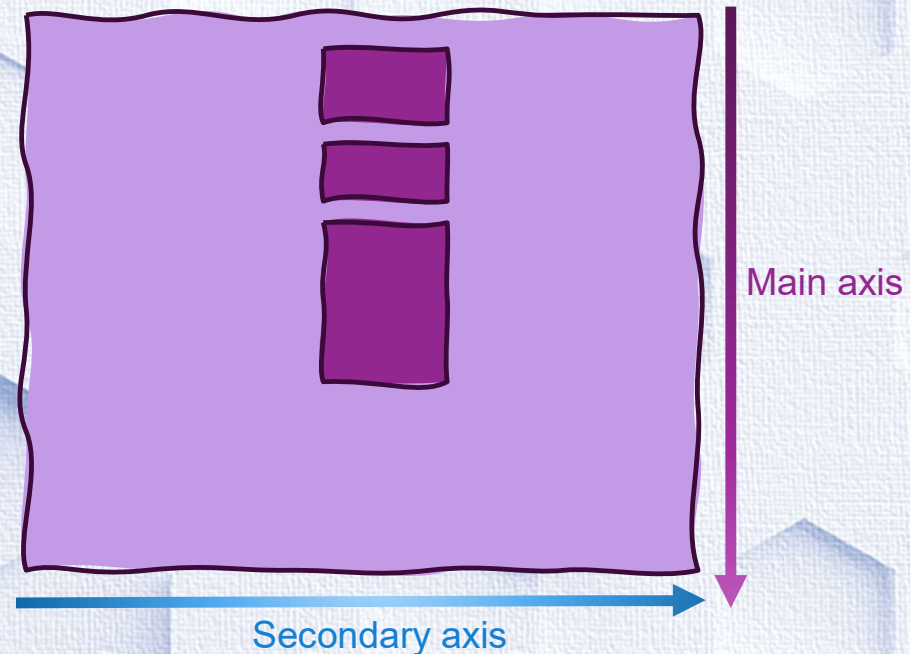
CSS



Main Axis and Secondary Axis

- The first Decision in flex container is **direction**
 - The **flex-direction** property
 - Vertical / Horizontal
 - Forward / Reverse
- Then comes **alignment**
 - **justify-content** property controls alignment on the main axis
 - **align-items** property controls alignment on the secondary axis

```
flex-direction: column;  
justify-content: flex-start  
align-items: center
```



Sass

ScSS

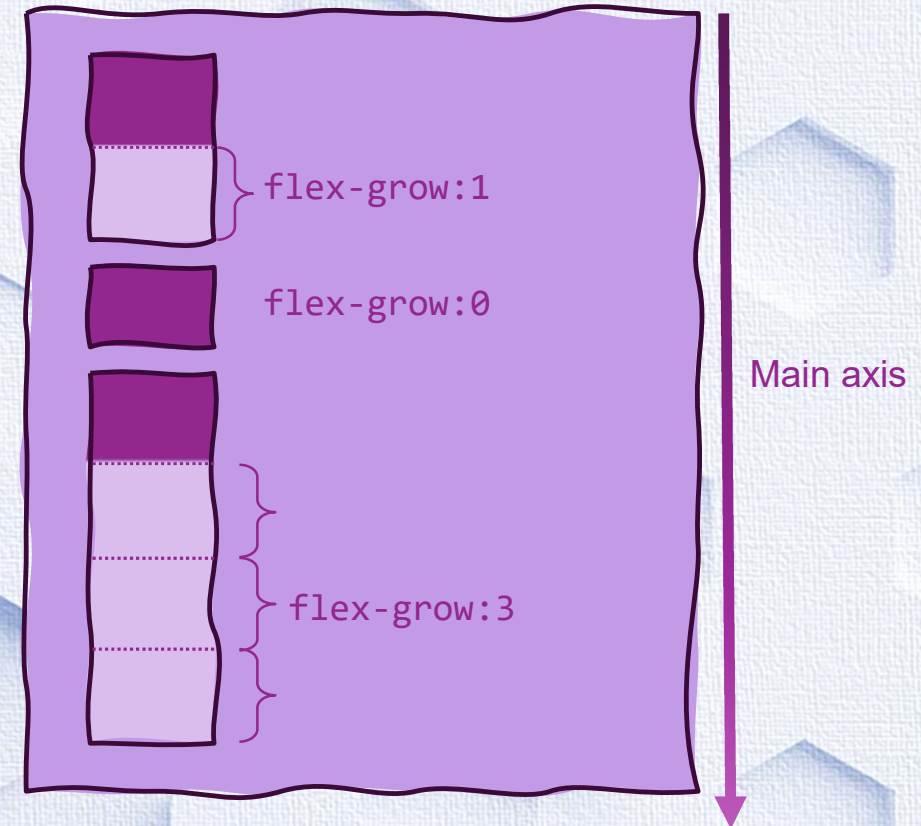
+

CSS



Space Distribution

- Only along the **main axis**
- If there is extra space:
 - **flex-grow** sets the relative share of a child item
- If there is missing space:
 - **flex-shrink** sets the relative share of a child item



ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



ScSS

+

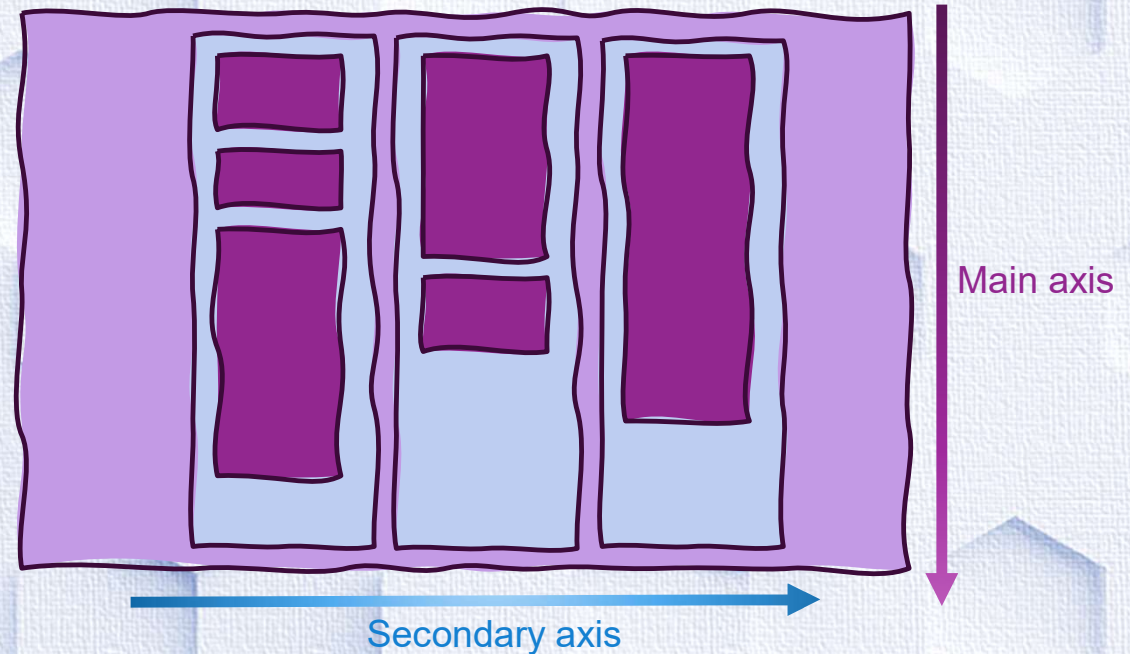
CSS



Wrapping

- **flex-wrap**: sets wrapping
- **align-content**: aligns the stacks
 - Along the secondary axis
- **wrap-reverse**
 - Reverses the direction of the stacks
 - Along the secondary axis

```
justify-content: flex-start;  
flex-wrap: wrap;  
align-content: center;
```



ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



ScSS

+

CSS



More features

- **order**

- Child property
- Can change the order of stacking

- **gap**

- Parent property
- Sets a common margin between elements
- Better than using **margin...**

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

Sass

ScSS

+

CSS



CSS Tricks

Grids

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

Sass

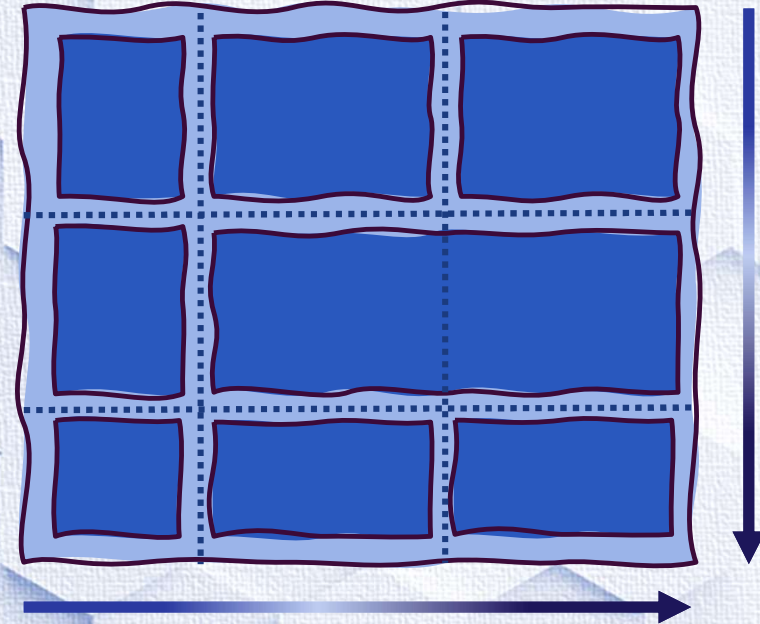
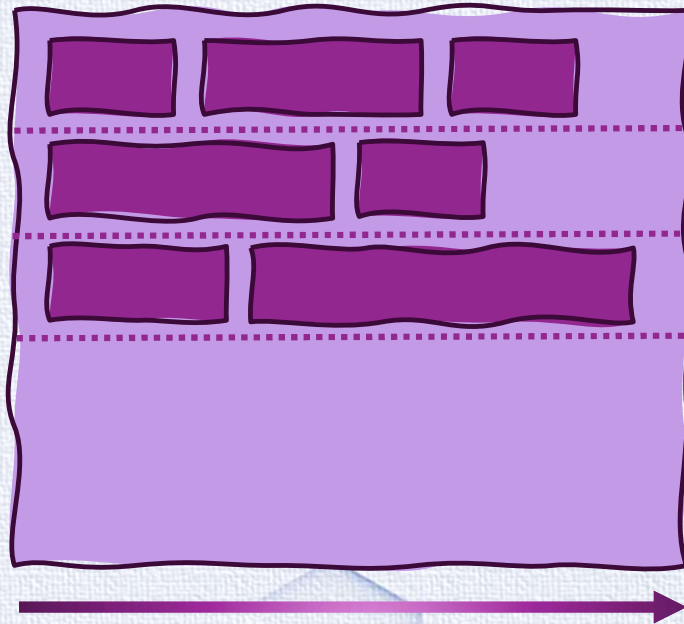
ScSS + CSS



Flex

vs

Grid



ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



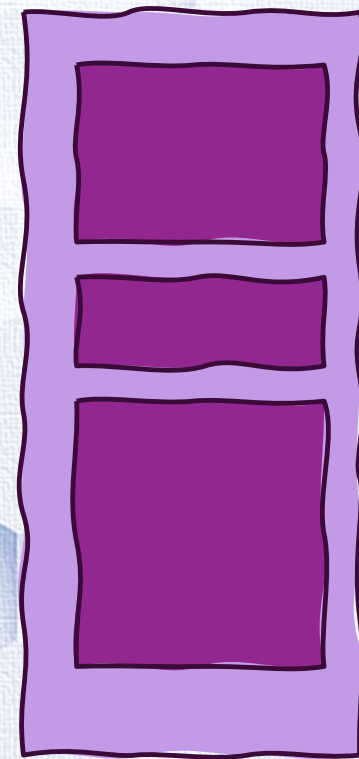
ScSS + CSS



What is flex?

- Flex is a **layout** that aligns and distributes **items** in a **container**.
- Enables flexible **spacing** and **resizing** of child elements.
- Supports **responsive** layouts with minimal CSS code.

display: flex



ORACLE

JOHN BRYCE
תלמדו היטב. זה עובד!
a matrix company



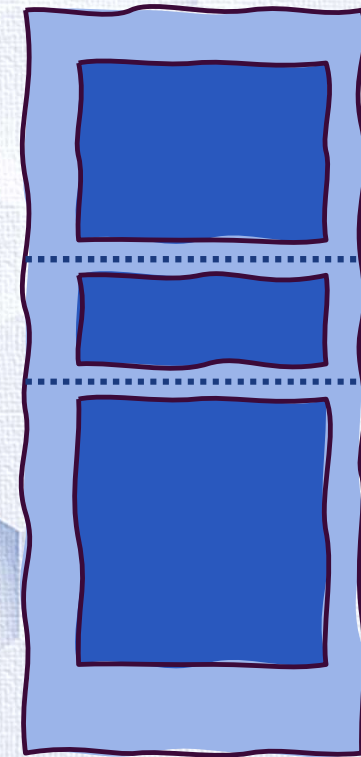
ScSS + CSS



What is grid?

- Grid is a **layout** that aligns and distributes **items** in a **container**.
- Enables gridable **spacing** and **resizing** of child elements.
- Supports **responsive** layouts with minimal CSS code.

`display: grid`



ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



ScSS

+

CSS



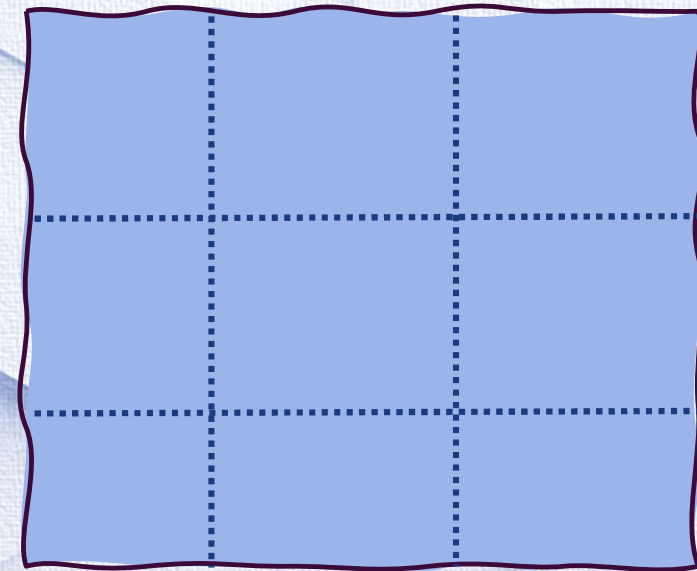
Grid Template

- The first Decision in grid container is **template**

- `grid-template-rows`
- `grid-template-columns`

- You can measure in
 - Pixels, points, ems, vh, etc etc
 - Percentage
 - **auto** (size by content)
 - **fr** (unit specific from grids dividing the remainder space by weight)
- Use the **repeat** function to create many many tracks

```
grid-template-rows: auto 100px 1fr  
grid-template-columns: repeat(3, 33%)
```



ORACLE

JOHN BRYCE
תלמדו היטב. זה עובד!
a matrix company



ScSS

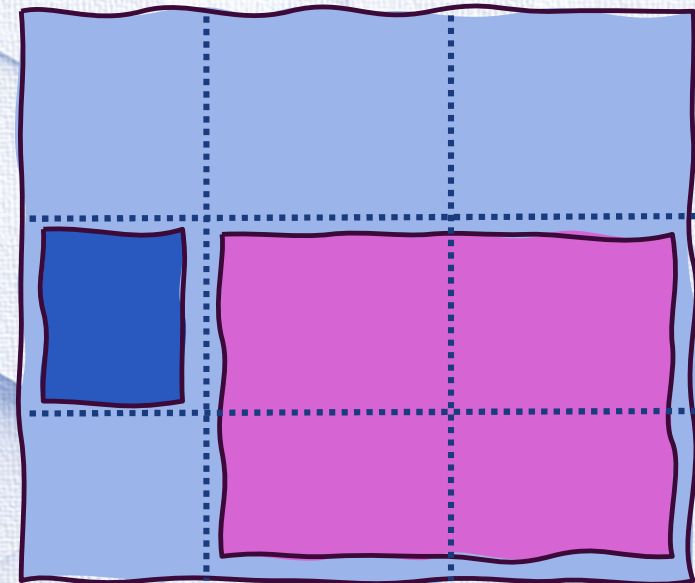
+

CSS



Item placements

- Automatic: Each child item takes the next free cell
- Automatic with constraint:
 - `grid-column: 1`
 - Take the next free cell that is on the first column
- Manual
 - `grid-column: 1`
 - `grid-row: 2`
- Items can span more than one cell
 - `grid-column: 2/4`
 - `grid-row: 2/4`



ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

Sass

ScSS

+

CSS



Line identifier

- Lines have many identifiers
 - Positive index
 - Negative index
 - Custom names



```
grid-template-rows: auto [menu] 100px [logo] 1fr  
grid-template-columns: 1fr [start] 1fr [finish] 1fr
```

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

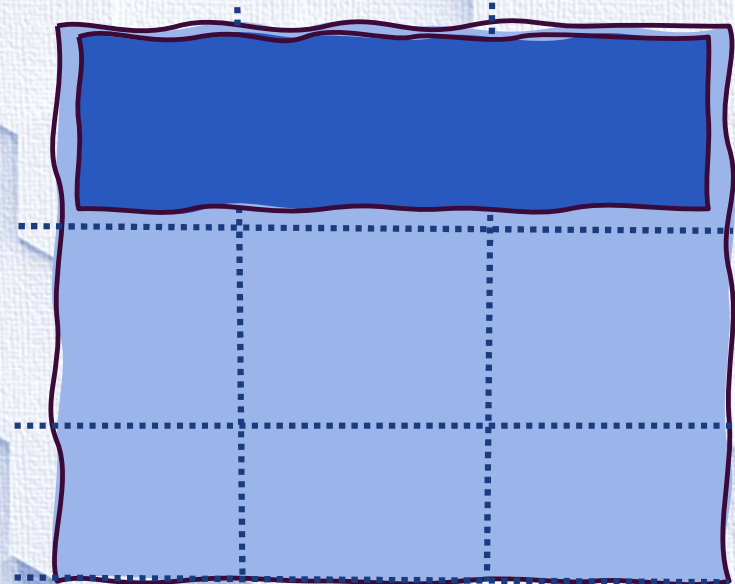
Sass

ScSS + CSS



Area identifier

- Use **grid-template-areas** (container) to define names for cells
- Use **grid-area** (child) to place items



```
grid-template-areas: 'header header header'
                    'menu      .      side'
                    'menu      footer footer'
```

```
grid-area: header
```

ORACLE

JOHN BRYCE
תלמדו היטב. זה עובד!
a matrix company

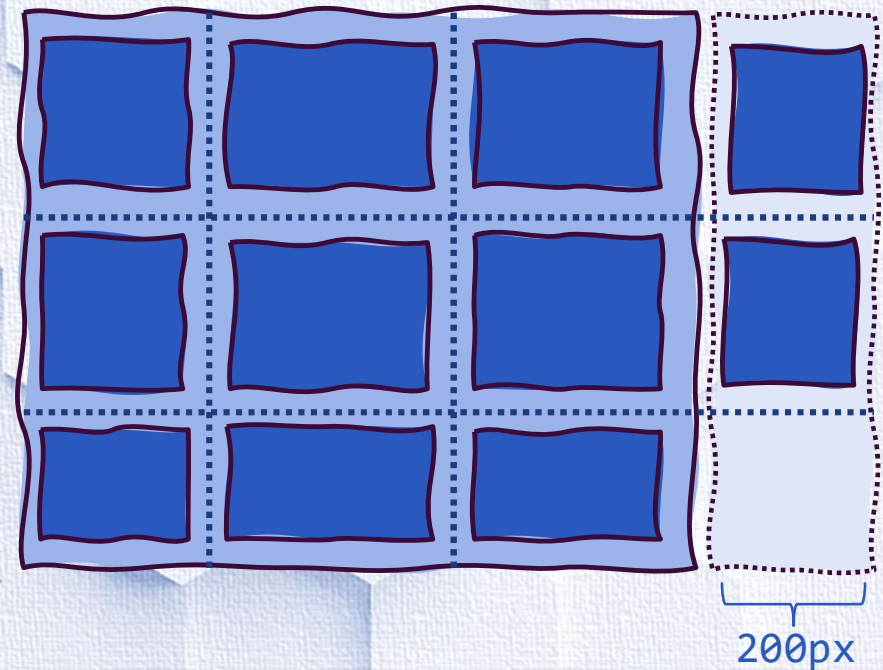
Sass

ScSS + CSS



Grids have direction too

- **grid-auto-flow** defines direction of auto placement
- It also defines direction of auto generation of tracks
- **grid-auto-rows** and **grid-auto-columns** define size of generated tracks
- The **dense** option defines if to save empty cells or fill them up



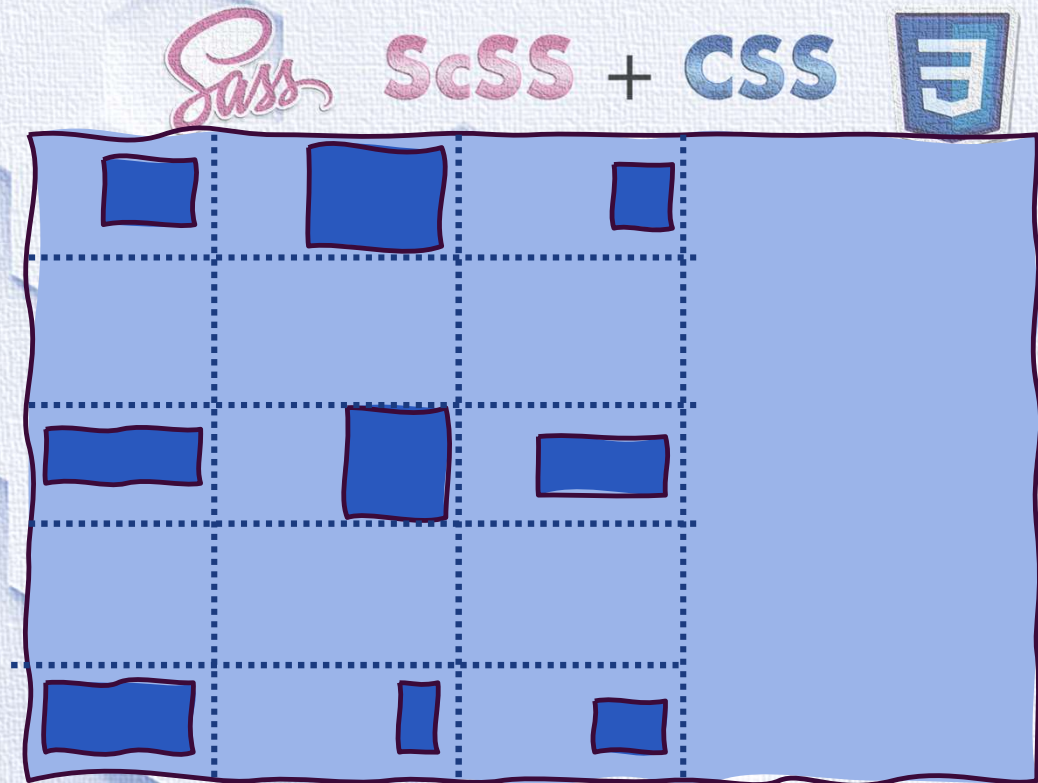
```
grid-auto-flow: column dense;  
grid-auto-column: 200px;
```

ORACLE

JOHN BRYCE
תלמדו היטב. זה עובד!
a matrix company

Grid alignment

- **justify-content** aligns tracks horizontally
- **align-content** aligns tracks vertically
- **justify-items** aligns items horizontally inside their tracks
- **align-items** aligns items vertically inside their tracks
- **align-self** and **justify-self** align one specific item inside its track



200px

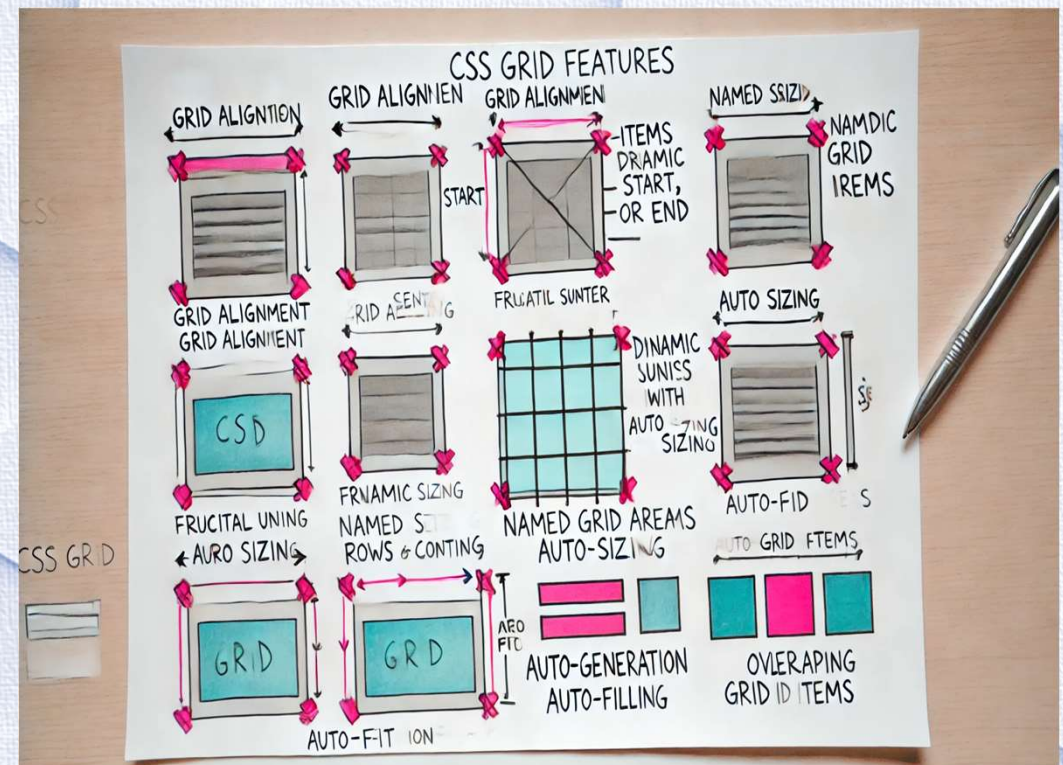
```
justify-content: start;  
align-content: space-between;  
justify-items: end;  
align-items: center;
```

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

More features

- **order**
 - Child property
 - Can change the order of placement
- **gap, column-gap, row-gap**
 - Parent property
 - Sets a common margin between elements
 - Better than using **margin...**
- **auto-fit, auto-fill**
 - Parent property
 - Fits tracks according to children and available space
 - Very useful for responsiveness
- Masonary
- Sub-grid



Sass

ScSS

+

CSS



CSS Custom Properties

A.K.A CSS vars

ORACLE

JOHN BRYCE
תלמדו היטב. זה עובד!
a matrix company

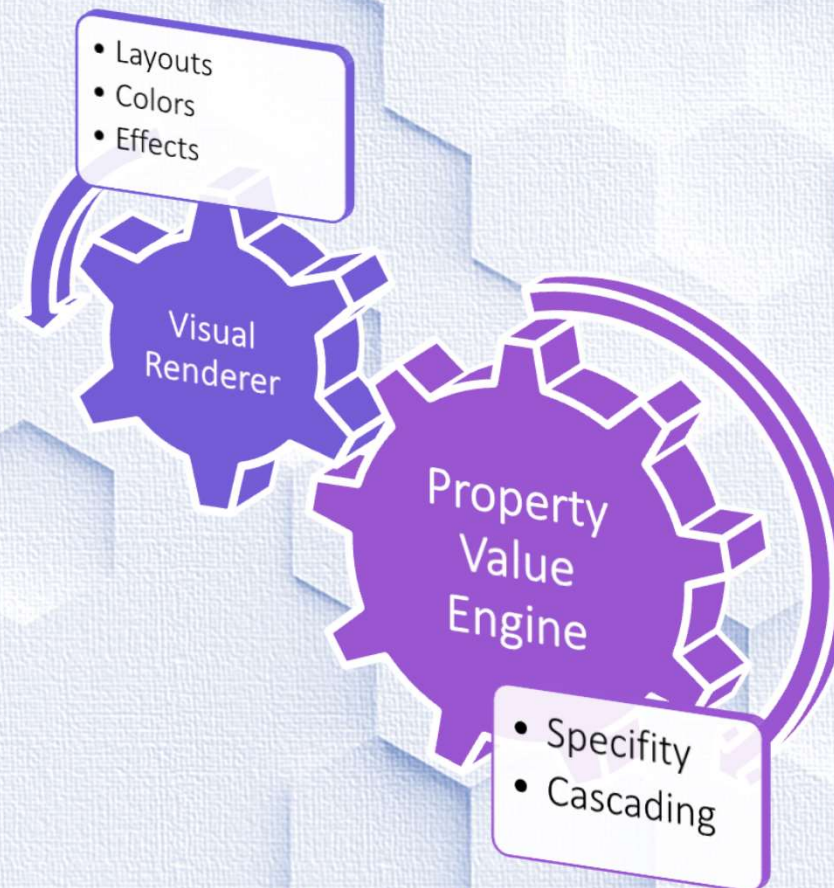
CSS is:

Sass

ScSS

+

CSS

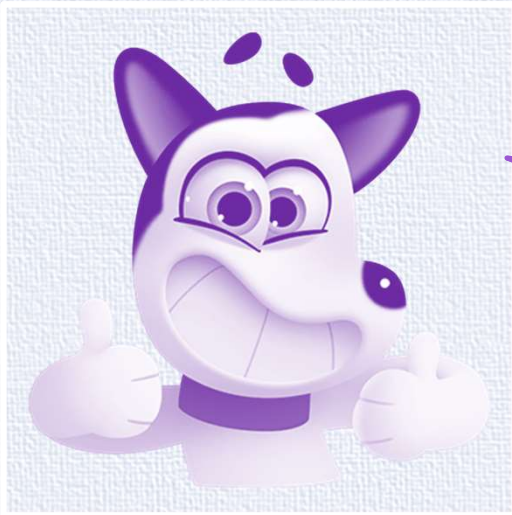


ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

Sass

ScSS + CSS



You can Cascade
properties that the
browser “understands”

You can Cascade
properties that you
make up



ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

Sass

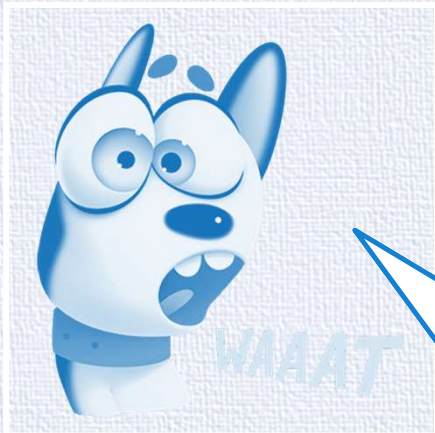
ScSS

+

CSS



Custom properties



Yea... but
what's the
point?

css-vars.css

```
/* Custom properties */  
:root {  
  --my-prop: 50;  
  --my-color: magenta;  
}
```

ORACLE

JOHN BRYCE
תלמדו היטב. זה עובד!
a matrix company



ScSS

+

CSS



The var function

- Copies from one property to another
- Can also have a fallback value
`var(--x, 42);`
- Can even use var as fallback value
`var(--x, var(--y, 42));`

css-vars.css

```
/* Custom properties */  
h1 {  
  color: var(--my-color);  
  text-decoration: var(--my-color) underline;  
}
```

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



ScSS

+

CSS



The calc function

- Performs calculations
- Can accept vars

calc.css

```
/* the calc function */  
h1 {  
  width: calc(var(--scale) * 20px);  
}
```

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



ScSS

+

CSS



Concatenation of unit types

units.css

```
body {  
  --value: 24;  
  --unit: px;  
  
  /* Nope */  
  font-size: var(--value) + var(--unit);  
  
  /* Yep */  
  font-size: calc(var(--value) * 1px);  
  
  /* Yep */  
  --pixel_converter: 1px;  
  font-size: calc(var(--value) * var(--pixel_converter));  
}
```

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

Sass

ScSS

+

CSS



The min, max, clamp functions

- Compare values
- Can mix units
- Can Use vars
- Very powerful for responsive!

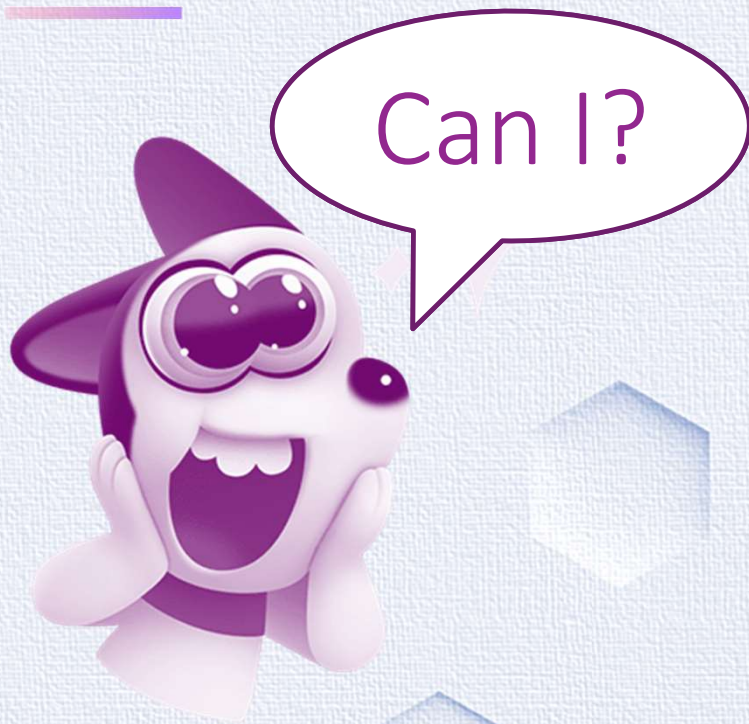
funcs.css

```
/* can mix different units */  
.panel {  
  width: min(100px, 50%);  
}  
  
/* can compare expressions */  
.box {  
  height: max(calc(var(--ratio) * 100px), 200px);  
}  
  
/* clamp combines min and max */  
div {  
  width: clamp(100px, 50%, 400px);  
}
```

ORACLE

JOHN BRYCE
תלמודו הייטק. זה עובד!
a matrix company

Can I use?



Sass

ScSS

+

CSS



CSS Variables (Custom Properties)

📄 - CR



Baseline

Widely available across major browsers

Permits the declaration and usage of cascading variables in stylesheets.

Usage

% of all users



Global

97.73% + 0.02% = 97.75%

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

You sure?



Sass

ScSS

+

CSS



Opera	IE	Chrom for Andro
0-34		
35		
6-113	6-10	
114	11	130

Well, you know...
Not on IE

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



ScSS

+

CSS



The @property rule

- All vars are by default
 - Inherited
 - Do not animate
 - Do not limit the value type
- You can define “strongly typed” vars using the `@property` rule
- And if you give them animatable type, they can animate

property-rule.css

```
@property --r {  
  syntax: '<angle>';  
  initial-value: 0deg;  
  inherits: false;  
}  
  
.star {  
  --r: 0deg;  
  transform: rotate(var(--r));  
  animation: spin 1s linear infinite;  
}  
  
@keyframes spin {  
  100% {  
    --r: 360deg;  
  }  
}
```

ORACLE

JOHN BRYCE
תלמדו היטב. זה עובד!
a matrix company

Sass

ScSS + CSS

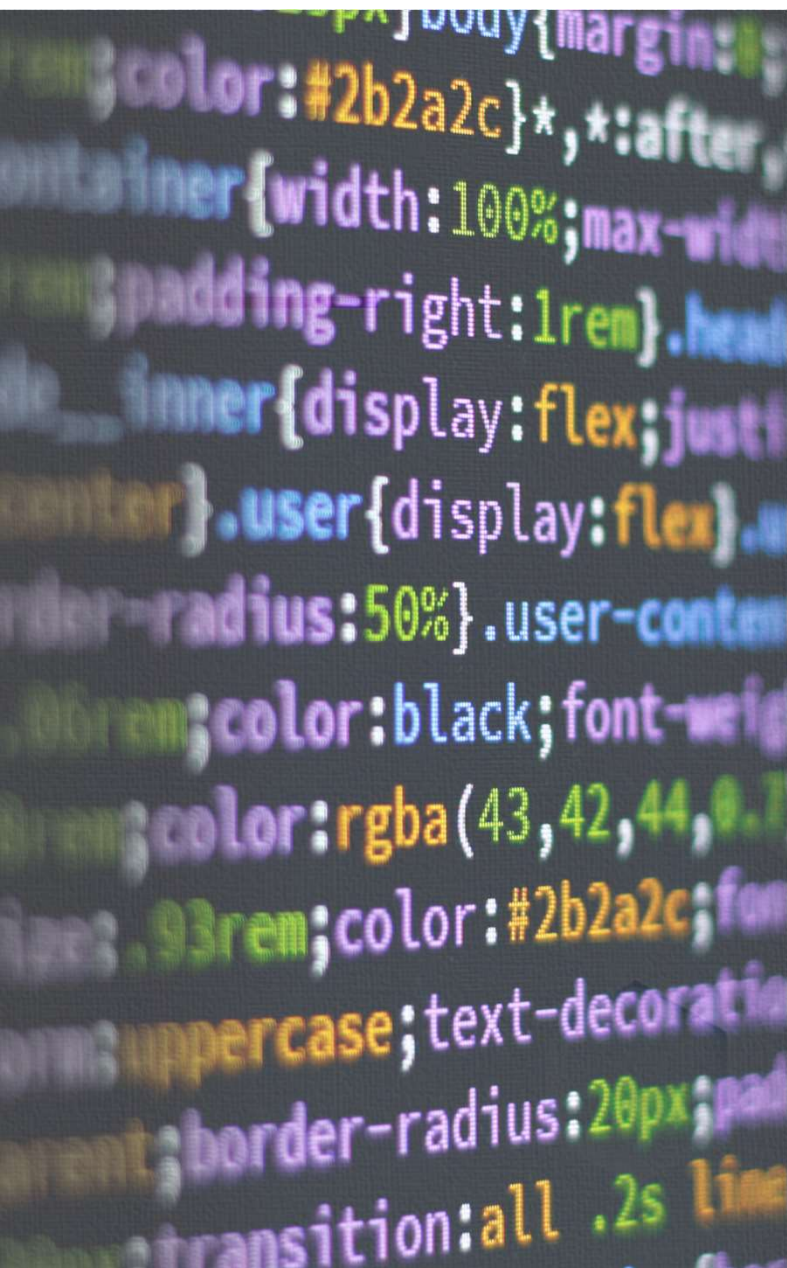


ScSS

Style Preprocessor

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



Sass

ScSS + CSS



What is a style preprocessor?

- A language that is more sophisticated than CSS
- Compiled into CSS
- ScSS Extends CSS.
 - You can Incrementally modify your code
 - You only need to learn the additions
- It's like Typescript and Javascript

ORACLE

JOHN BRYCE
תלמדו היטב. זה עובד!
a matrix company



ScSS

+

CSS



ScSS variables

- Have \$ sign prefix
- Can hold any value
- The value can be reassigned (changed)
- But they only exist in compile time...

variables.scss

```
$base-color: #c6538c;  
$border-dark: rgba($base-color, 0.88);  
  
.alert {  
  border: 1px solid $border-dark;  
}
```



calc.css

```
.alert {  
  border: 1px solid rgba(198, 83, 140, 0.88);  
}
```

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



ScSS

+

CSS



Functions in ScSS

- Start with **@function**
- Can receive parameters
- Use the “...” operator to receive arbitrary arguments (list)
- Use the flow control rules for conditions and loops
 - @each
 - @for
 - @if - @else
- Return value using **@return**

functions.scss

```
@function sum($numbers...) {  
  $sum: 0;  
  @each $number in $numbers {  
    $sum: $sum + $number;  
  }  
  @return $sum;  
}  
  
.micro {  
  width: sum(50px, 30px, 100px);  
}
```



functions.css

```
.micro {  
  width: 180px;  
}
```

ORACLE

JOHN BRYCE
תלמדו היטב. זה עובד!
a matrix company



ScSS

+

CSS



Interpolation

- Use interpolation to embed the result of a sass expression
 - Selectors
 - Property names
 - Property values
- Use the `#{}` syntax

```
interpolation.scss

$name: "border";

.icon-#{$name} {
  background-image: url("/icons/#{$name}.svg");
  position: absolute;
  #{$name}-bottom: 0;
  #{$name}-top: 0;
}
```



```
interpolation.css

.icon-border {
  background-image: url("/icons/border.svg");
  position: absolute;
  border-bottom: 0;
  border-top: 0;
}
```




ScSS

+

CSS



mixins

- Are like scss templates
- Do not confuse them with functions
 - Functions calculate a value
 - Mixins yield CSS
- Use the `@mixin` syntax
- Mixins can also accept parameters
- You call a mixin using `@include`.

mixin.scss

```
@mixin square($size, $radius: 0) {  
  width: $size;  
  height: $size;  
  @if $radius != 0 {  
    border-radius: $radius;  
  }  
}  
  
.avatar {  
  @include square(100px, $radius: 4px);  
}
```

mixin.css

```
.avatar {  
  width: 100px;  
  height: 100px;  
  border-radius: 4px;  
}
```

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

\$primary-color: #333;



partials

- You create partial file for reuse
- It can be incorporated into many CSS files
- But it should not be generated on its own
- They start with underscore as a convention
- They should not yield CSS. Only functions, variables and mixins

```
_partial.scss

$primary-color: #333;

@mixin square($size, $radius: 0) {
  width: $size;
  height: $size;
  @if $radius != 0 {
    border-radius: $radius;
  }
}
```



```
_partial.css
```




ScSS + CSS



Modularity with @use

- Old versions of ScSS use `@import`
- But it is deprecated now
- Instead use `@use`
- Loads the partial members into a namespace
- Can be aliased
- No need for the `_` prefix

use.scss

```
@use 'partial' as base;

.inverse {
  background-color: base.$primary-color;
  color: white;
}
```



use.css

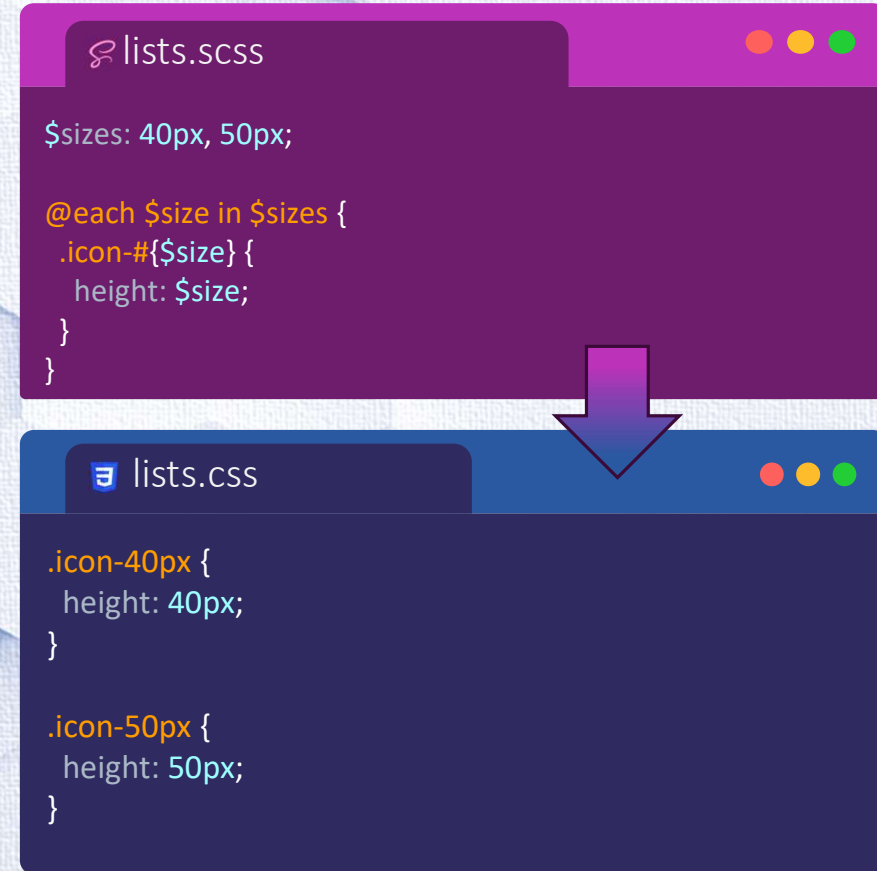
```
.inverse {
  background-color: #333;
  color: white;
}
```

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

Lists

- Can hold a list of values
- The built-in module **sass:list**
 - Contains list-oriented functions
 - **nth** fetches a value by index
 - **append** adds a value
 - **index** returns the index of an item
 - And many more...
- Loop using **@each \$item in \$list**
- Are immutable!





ScSS

+

CSS



Maps

- Are lists of key-value pairs
- The built-in module **sass:map**
 - Contains map-oriented functions
 - **get** fetches a value by key
 - **set** sets a value by key
 - **merge** returns the index of an item
 - And many more...
- Loop using **@each \$key, \$item in \$map**
- Are immutable!

maps.scss

```
$icons: ("eye": "\f112", "start": "\f12e");

@each $name, $glyph in $icons {
  .icon-#{$name}:before {
    font-family: "Icon Font";
    content: $glyph;
  }
}
```



maps.css

```
.icon-eye:before {
  font-family: "Icon Font";
  content: "\f112";
}

.icon-start:before {
  font-family: "Icon Font";
  content: "\f12e";
}
```


The Sass logo, featuring the word "Sass" in a stylized, cursive pink font.

ScSS

+

CSS



Color Module

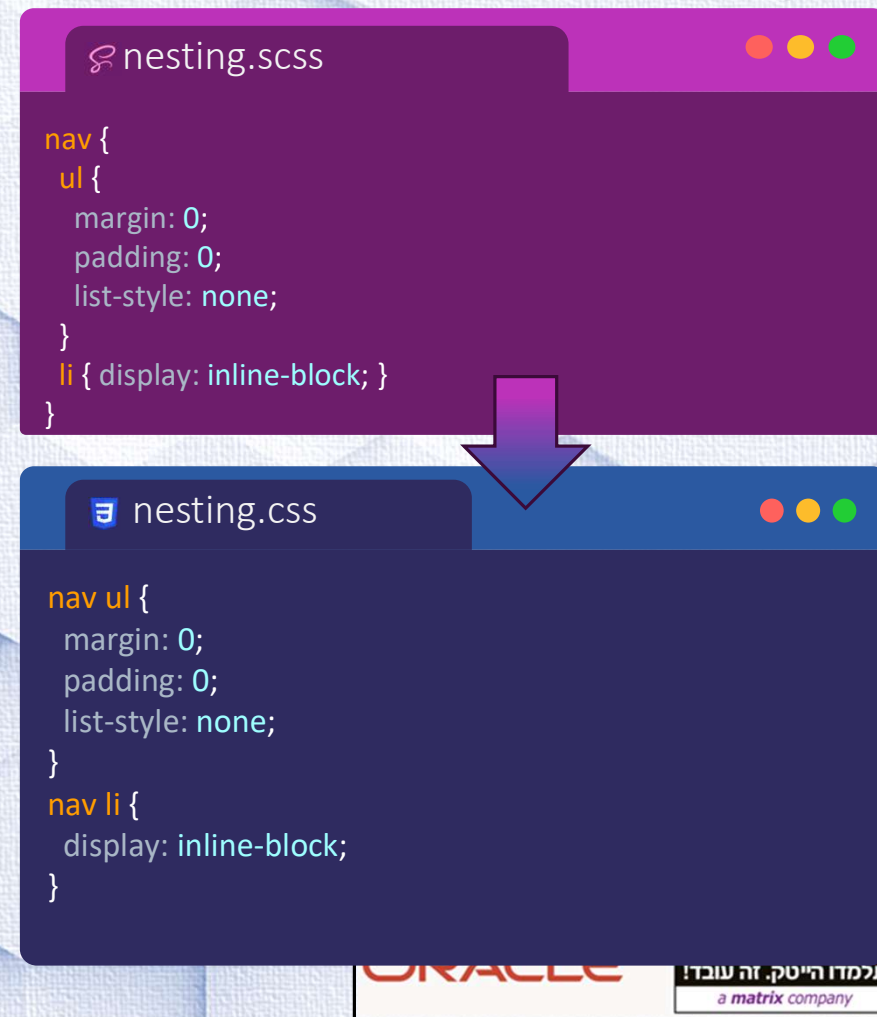
- Can perform complex color calculations
 - Mix colors
 - Adjust rgb and hsl channels
 - Convert from one system to another
 - Transfer to grayscale
 - Invert, Complement,
 - But only in Compile time ☹
- There is now also color-mix
 - in CSS!
 - At runtime!!!
 - Yea, really!

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

Nesting

- In ScSS you can nest rules inside rules
- This makes your ScSS file Hierarchical
- Saves from using long list of descendants on each rule





ScSS

+

CSS



The parent selector &

- Is a special ScSS selector
- It represent the current parent selector
- Can be used to combine selectors in nested code
- Can be used as variable
- Can be tested in @if statements to behave differently at root level and inside parent

parent.scss

```
.alert {  
  &:hover {  
    font-weight: bold;  
  }  
  :not(&) {  
    opacity: 0.8;  
  }  
  &-dark {  
    color: white;  
  }  
}
```



parent.css

```
.alert:hover { font-weight: bold; }  
:not(.alert) { opacity: 0.8; }  
.alert-dark { color: white; }
```

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

Sass

ScSS

+

CSS



Themeing Apps

Using ScSS and CSS

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company



Sass

ScSS + CSS



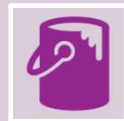
Theming



We used to calculate a lot of colors in ScSS



Custom properties: calculate properties instead and use them in CSS



color-mix: rely on one single custom property and variations that are calculated from it

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

**ORACLE
TECH DAYS**

ORACLE

JOHN BRYCE
תלמדו הייטק. זה עובד!
a matrix company

Thank You
