Telerik Academy

Stylus Overview

Expressive, dynamic, robust CSS

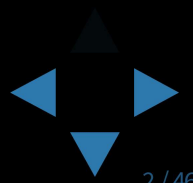
CSS Styling

Telerik Software Academy

telerikacademy.com

Telerik Academy

Table of Contents



Telerik Academy

Table of Contents

Telerik Academy

Preprocessors Overview

What is a Preprocessor?

Telerik Academy

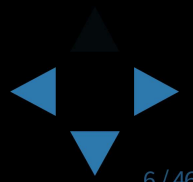
Preprocessors Overview

- A preprocessor is a program that processes its input data to produce output that is used as input to another program
  - Sass, Less, Stylus are preprocessors for CSS
  - CoffeeScript, TypeScript are preprocessors for JavaScript
  - Jade, Ejs are preprocessors for HTML

Telerik Academy

Stylus Basics

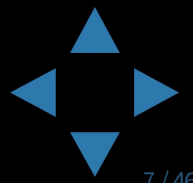
What is Stylus?



Telerik Academy

What is Stylus?

- Stylus is a preprocessor for CSS
  - All Stylus code is compiled to pure CSS
  - Extends CSS, without breaking it
    - Adds a lot of syntactic sugar
    - Adds a "dynamic" functionality
    - Removes "not-necessary" code



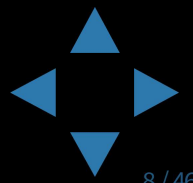
Telerik Academy

Stylus Syntax

- The Stylus syntax is clean and easy
  - Removes all not-necessary symbols
    - Semicolons, curly brackets, etc…
  - Significant whitespace is what matters
    - Indentation marks the scope

```
/* CSS */
#root{
  display: block;
  width: 960px;
  margin: 0 auto;
  border: 1px solid black;
}
```

```
/* stylus */
#root
  display block
  width 960px
  margin 0 auto
  border 1px solid black
```

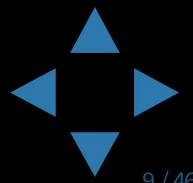


Telerik Academy

Selectors in Stylus

- Stylus selectors are absolutely the same as in regular CSS
  - Every CSS code is valid Stylus code!
  - `.class`, `#id`, `tagName`
- *Examples:*

```
#root .nav
  list-style-type none
#root .nav .nav-item
  float left
  margin 0
  padding 0
```



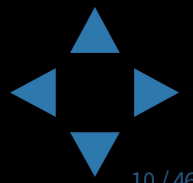
Telerik Academy

Selectors in Stylus

- Yet, Stylus supports nesting of selectors:

```
/* Stylus */
.hlist
  list-style-type none
  .list-item
    float left
    border-right 1px solid #ccc
    &:last-of-type
      border-right none
```

```
/* CSS */
.hlist {
  list-style-type: none;
}
.hlist .list-item {
  float: left;
  border-right: 1px solid #ccc;
}
.hlist .list-item:last-of-type {
  border-right: none;
}
```

Telerik Academy

Stylus selectors

Nesting selectors

Demo

Telerik Academy

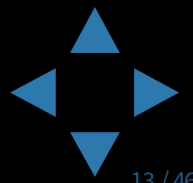
Stylus: Setup and Usage

Setting up Node.js, Stylus and coding environment

Telerik Academy

Stylus: Setup and Usage

- To use Stylus:
  - Install Node.js
  - Install Stylus package for Node.js
  - Install a plugin for your editor



Telerik Academy

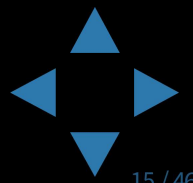
Installing Node.js

- Visit the Node.js website: https://nodejs.org/
- Download and install Node.js
- Make sure Node.js is added to PATH
  - Adding programs to PATH under Windows
- Node.js is installed on the machine and can be used through the CMD/Terminal

Telerik Academy

Adding Node.js to Path on Linux and OS X

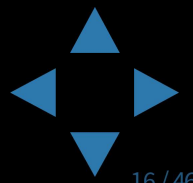
- Open `~/.bashrc` with `admin sudo`
- Find the Node.js path
  - Usually it is `/usr/bin/node`
- Add the Node.js path to `~/.bashrc`
- You have Node.js in the PATH



Telerik Academy

Installing Stylus

- Install Node.js
  - Make sure Node.js is in PATH
- Open CMD/Terminal
- Run the following line:
  - `npm install -g stylus`
  - this will install Stylus globally on the machine



Telerik Academy

Using Stylus

- Stylus code must be compiled to CSS
- There are a few ways to achieve that:
  - Using the Node.js Terminal/CMD
    - Stylus command line docs
  - Or use plugins for your favorite editor:
    - Sublime Text 2/3
    - Atom.io
    - Visual Studio
    - Visual Studio Code
    - Brackets
    - WebStorm

Telerik Academy

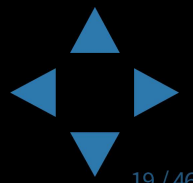
Stylus Features

Variables, Functions, Mixins, Variables, etc…

Telerik Academy

Stylus Features

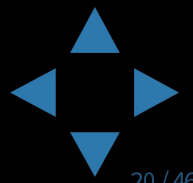
- Stylus supports:
  - Variables
  - Interpolation
  - Property look-up
  - Functions and mixins
  - Selector inheritance
  - Conditionals and Iteration
  - Many more



Telerik Academy

Stylus Features: Variables

- Variables in Stylus are defined easy:
  - As a regular property/value pair, but using "=" (equals sign) instead of " : " (colons)
  - Can be used to store colors, size, etc…
- Usable to set default background-color, font-color, font-size, etc…

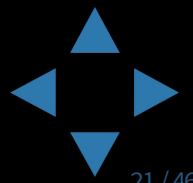


Telerik Academy

Stylus Features: Variables

```
/* Stylus */
link-color = #ffffff;
v-link-color = #646363;
a
  color link-color
  &:visited
    color @v-link-color
```

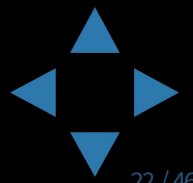
```
/* CSS */
a {
  color: #fff;
}
a:visited {
  color: #646363;
}
```



Telerik Academy

Stylus Features: Variables

Demo



Telerik Academy

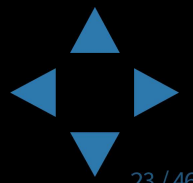
Stylus Features: Interpolation

- Interpolation means inserting variable value as a property/selector name
  - Interpolation is performed using curly brackets {...}
- Stylus

```
side = 'top'
number = 5
.box-{number}
    border-{side} 1px solid black
```

- CSS

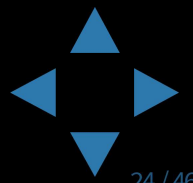
```
.box-5 {
  border-top: 1px solid #000;
}
```

Telerik Academy

Stylus Features: Interpolation

Demo



Telerik Academy Stylus Features: Property Look-up

- Stylus supports property look-up
  - i.e. check the values of a property
- Stylus

```
.box
    width 100px
    height (@width/2)
    border-radius max(@width, @height)
```

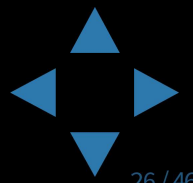
- CSS

```
.box {
  width: 100px;
  height: 50px;
  border-radius: 100px;
}
```

Telerik Academy

Stylus Features: Property Look-up

Demo



Telerik Academy

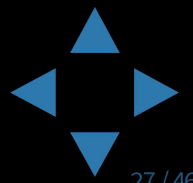
Stylus Features: Built-in Functions

- Stylus has a set of predefined functions for:
  - Working with colors:

```
color = /* some color */
dark(color) /* returns "true", if "color" is dark */
light(color) /* returns "true" if "color" is light */
```

```
color: darken(#123456, 10%) /* returns #102f4d */
color: lighten(#123456, 15%) /* returns #1d5288 */
```

- Find all built-in functions in Stylus at https://learnboost.github.io/stylus/docs/bifs.html

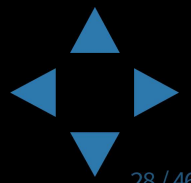


Telerik Academy

Stylus Features: Built-in Functions

- Working with lists:

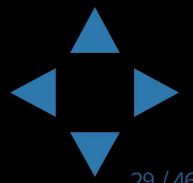
```
props = (padding 5px) (display block)
push(props, (float left))
```

Telerik Academy

Stylus Features: Built-In Functions

Demo



Telerik Academy

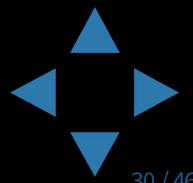
Stylus Features: Functions

- Stylus supports definition of developer functions:

```
sum(items)
    sum = 0px
    for val in items
        sum = sum + val
    sum /* return sum */
```

- Used as a regular function

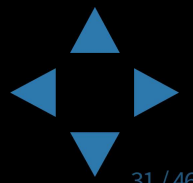
```
font-size: sum(14px 28px 45px)
height: sum(@width  + 150px)
```

Telerik Academy

Stylus Features: Mixins

- Both mixins and functions are defined in the same manner
  - Yet they are applied in different ways
    - Functions return a value, mixins generate CSS



Telerik Academy

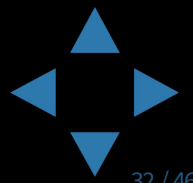
Stylus Features: Mixins

- Stylus

```
vendor(prop, args)
    -webkit-{prop} args
    -moz-{prop} args
    {prop} args
.button
  vendor('opacity', 0.5)
  vendor('border-radius', 15px)
```

- CSS

```
.button {
  -webkit-opacity: 0.5;
  -moz-opacity: 0.5;
  opacity: 0.5;
  -webkit-border-radius: 15px;
  -moz-border-radius: 15px;
  border-radius: 15px;
}
```

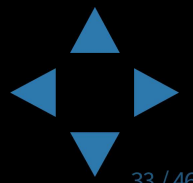
Telerik Academy

Mixins

- How to define mixins?
  - Specify a mixin name
  - Write styles in them - normal Stylus
  - How to use the mixin?
    - Write the name of the mixin and it will get replaced by the Stylus code

```
clearfix
  zoom 1
  &:after
    /* code for clearfix */
```

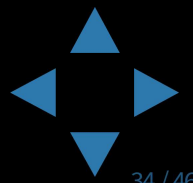
```
ul#main-nav{
  clearfix()
}
```

Telerik Academy

Stylus Features: Mixins

Demo



Telerik Academy Stylus Features: Mixins with Arguments

- Mixins can also be defined with parameters
  - i.e. for gradient-background
- Use the arguments like a C#/JS/C++ methods

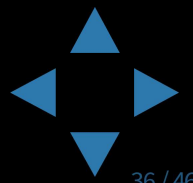
```
box(border = none, bg = rgba(0, 0, 0, 0.7), size = 200px)
    width size
    height size
    border border
    background bg
    border-radius 15px

.rect
  display inline-block
  box '1px solid black' rgba(123, 0, 0, 0.6) 100px
```

Telerik Academy

Selector Inheritance

- Selectors in Stylus can be inherited using `@extend`
  - Meaning add a selector to a previously defined selectors list:
- Stylus

```
.clearfix
  zoom 1
  &:after
    /* code for clearfix */
.list-item
  float left
  padding 5px 15px
  &:last-of-type
    border-right: none
    @extend .clearfix
```

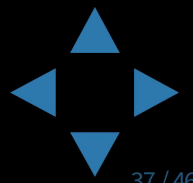



Telerik Academy

Selector Inheritance

- CSS

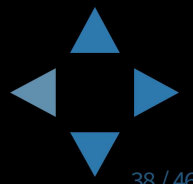
```
.clearfix,
.list-item:last-of-type {
  zoom: 1;
}
.clearfix:after,
.list-item:last-of-type:after {
  /* code for clearfix */
}
.list-item { /* … */ }
.list-item:last-of-type { /* … */ }
```

Telerik Academy

Stylus Features: Selector Inheritance

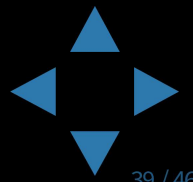
Demo



Telerik Academy

@extend or Mixins?

- Why both `mixins` and `@extend`?
  - `Mixins` have parameters and generate different code, based on the parameters
  - `@extend` provides a way to minify the number of classes on an element
    - Instead of adding `.clearfix` to all elements that clear, add it to the selector that the elements already have



Telerik Academy

Stylus Features: Mixins with Arguments

Demo

Telerik Academy

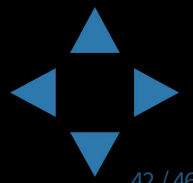
Stylus: Scripting, Libraries and Importing

Telerik Academy

Stylus Scripting

- Stylus supports elementary scripting
  - Something like JavaScript, but not quite
  - Conditionals:

```
body
  if theme is 'dark'
    background #333
    color #ccc
  else if theme is 'light'
    background #ccc
    color #333
  else
    error('unknown theme!')
```

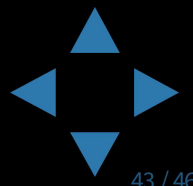


Telerik Academy

Stylus Scripting

- Conditionals:

```
for i in (1...15)
    .item-{i}
        if i % 2
            c = yellowgreen
            bc = #ccc
        else
            c = lightblue
            bc = #333
        color c
        background-color bc
```



Telerik Academy

Creating Libraries and Importing Stylus files

- Like CSS, Stylus has a `@import` directive
  - The CSS loads another CSS file
  - In CSS, `@import` loads a CSS file with an additional HTTP request
  - In Stylus, `@import` just loads the code from the file inside our file

```
.clearfix
  /* … */
linear-gradient(stops)
  /* … */
```

```
@import 'utils.styl'

body
  linear-gradient #333 #fff
```