Full-stack Application Development

using ReactJS, Express and TypeScript

The Course Main Goals

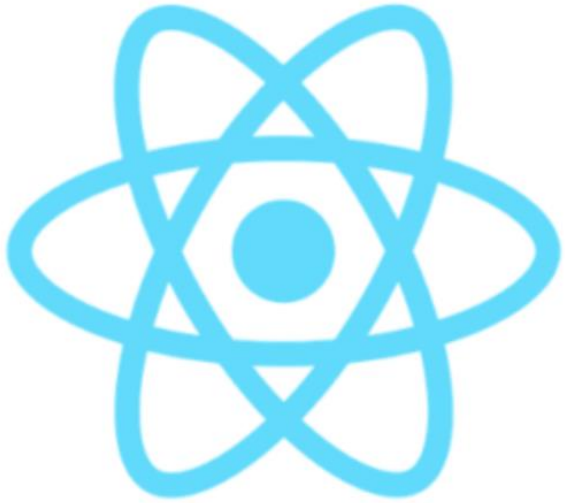
The course provides in-depth study of state-of-the-art *TypeScript*, *ReactJS*, *NodeJS*, and *Express* platform for rapid development of modern, component-based, mobile-first, responsive full-stack web applications, that are easy to extend and maintain.

Time Schedule

- 9.00 – 10.30 – Block 1
- 10.30 – 10.45 – Break
- 10.45 – 12.15 – Block 2
- 12.15 – 13.00 – Lunch break
- 13.00 – 14.30 – Block 3
- 14.30 – 14.45 – Break
- 14.45 – 16.15 – Block 4

Where to Find The Code and Materials?

https://github.com/iproduct/fullstack-typescript-react



Why to Attend This Training?

What will you learn?

You Will Learn How To:

- Set up different kinds of TypeScript projects – front-end with React + Redux, and back-end with NodeJS and Express;
- Integrate them in a full-stack project using REST/JSON APIs;
- Build type-safe components, single page apps, and backend services;
- Implement real-time server-push event streaming using Socket.IO;
- Understand how to boost development performance with TypeScript, and different tooling alternatives;
- Implement unit, integration and functional (UI, E2E) tests with Jest, Enzyme, and TestCafe;
- Debug full-stack applications written in TypeScript

Course Schedule

- [08.06, 09.06] - Introduction to HTML5, CSS and JavaScript – 12h
- [10.06] - HTTP Client and Fetch APIs – 6 h
- [11.06] - Novelties in ES6+ – 6 h
- [12.06] - TypeScript – 6h
- [15.06] - Single Page Applications (SPA) development with ReactJS – 2h. Building React components and applications with TypeScript – 6h
- [16.06] - TSX in depth – 4h
- [17.06] - Working with forms – 6h
- [18.01] - Components composition in depth – 6h

Course Schedule

- [19.06] - New lifecycle methods – 3h. Building purely functional components using new Hooks API – 6 h
- [22.06] - Other novelties in React 16+ – 3h
- [23.06] - Dependency Injection (DI) of custom application services using React Context – 4 h
- [24.06] - SPA routing using React Router v5 – 8 h
- [25.06] - Testing React Components – unit and end-to-end (E2E, UI) testing using – 6h
- [26.06] - Using Component Libraries – 6h

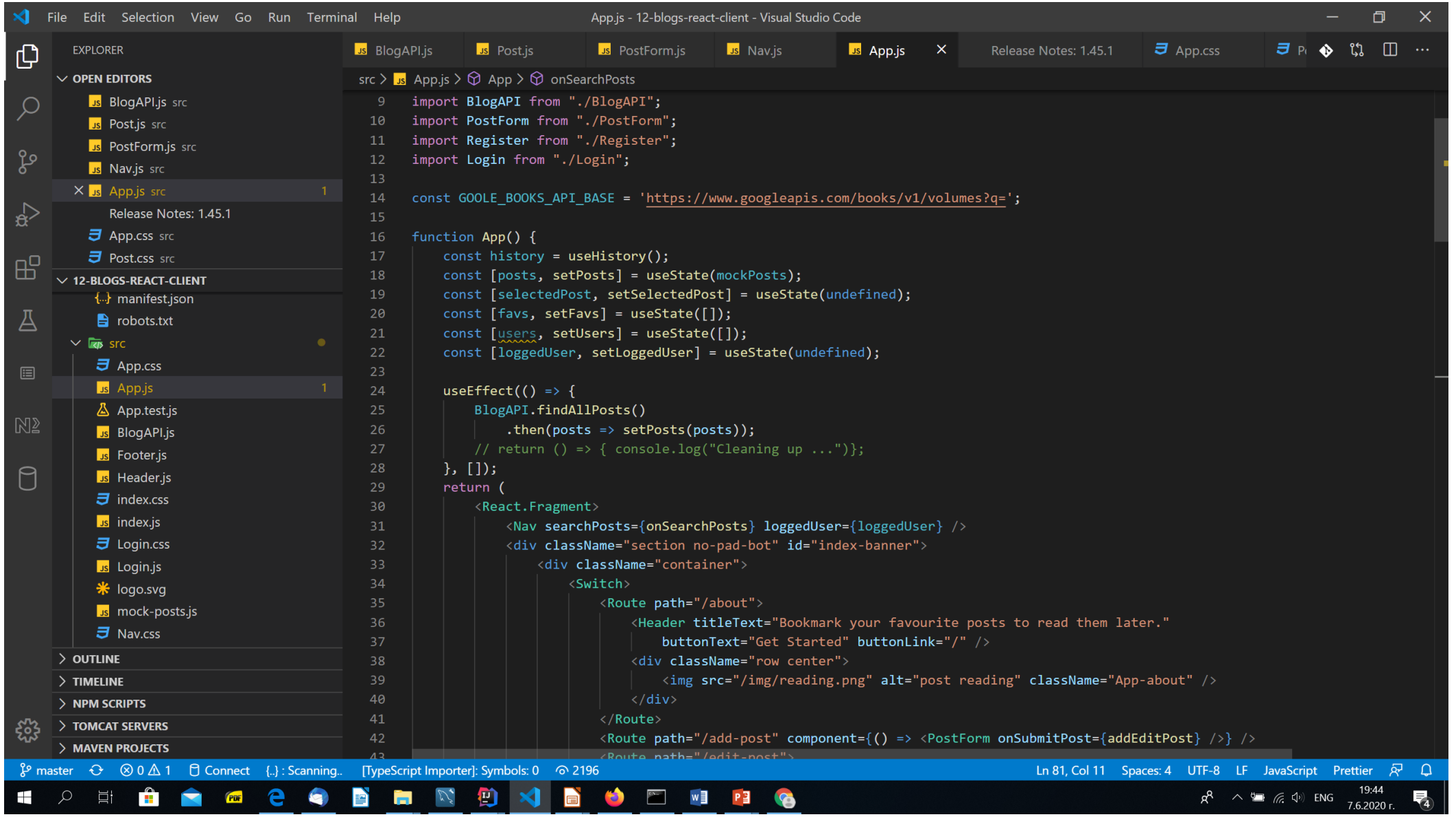
Course Schedule

- [29.06] - Redux – predictable state container for JavaScript apps – 8h
- [30.06] - Server side application development with Node.js and TypeScript – 8h
- [01.07, 02.07, 03.07] - REST service API development using Express.js. Database integration: document and relational DBs – MongoDB and MySQL. – 14h
- [09.07] - Final projects demonstration: presentation + demonstration – 10 min per project

Software to Install

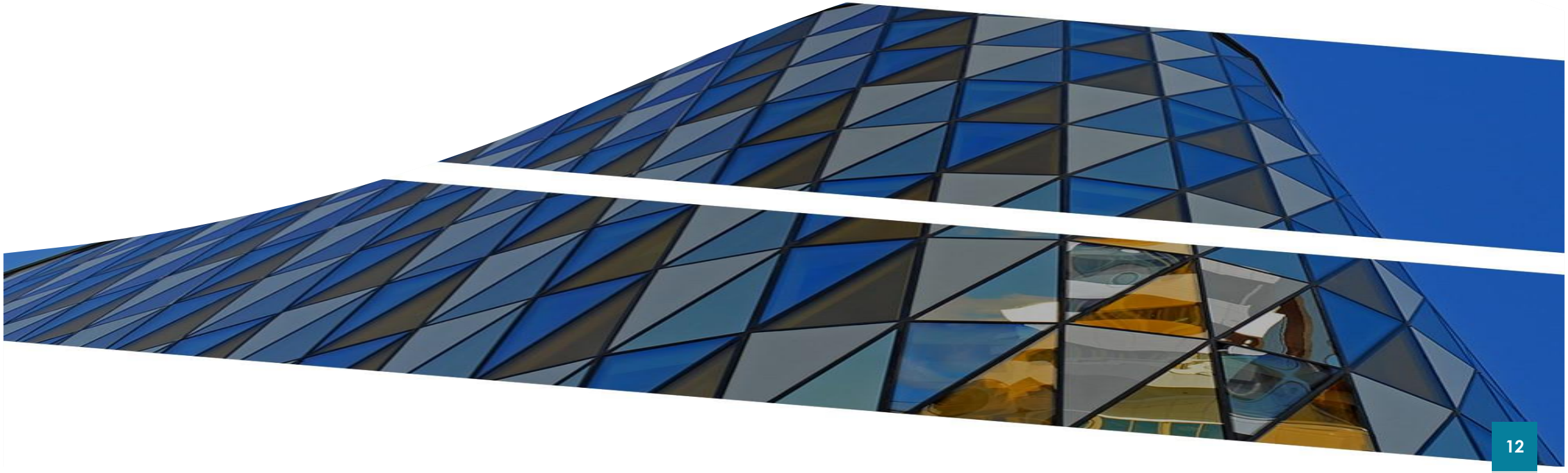
- NodeJS - https://nodejs.org/en/download/current/
- Visual Studio Code - https://code.visualstudio.com/download
- MongoDB Community Server - https://www.mongodb.com/try/download/community
  Don't forget to install MongoDB Compass (GUI) too.
- MySQL Community Server v8 - https://dev.mysql.com/downloads/mysql/
- MySQL Workbench - https://dev.mysql.com/downloads/workbench/
- Yarn package manager for NodeJS - https://classic.yarnpkg.com/en/docs/install/#windows-stable
- Chrome Web Browser

Visual Studio Code



Introduction to HTML5, CSS & JavaScript

Web design technologies recap



Introduction to HTML 5

Base tags, semantic markup, styling

We will follow the HTML 5 Tutorials at W3Schools:

- HTML Elements
- Styles and formatting
- Integration with CSS
- Basic building blocks: links, images, tables, lists
- Block and inline elements
- Ids and classes. Basic CSS selectors
- Adding behaviour with JavaScript
- Basic layouts with HTML and CSS
- Semantic markup

We will follow the HTML 5 Tutorials at W3Schools:

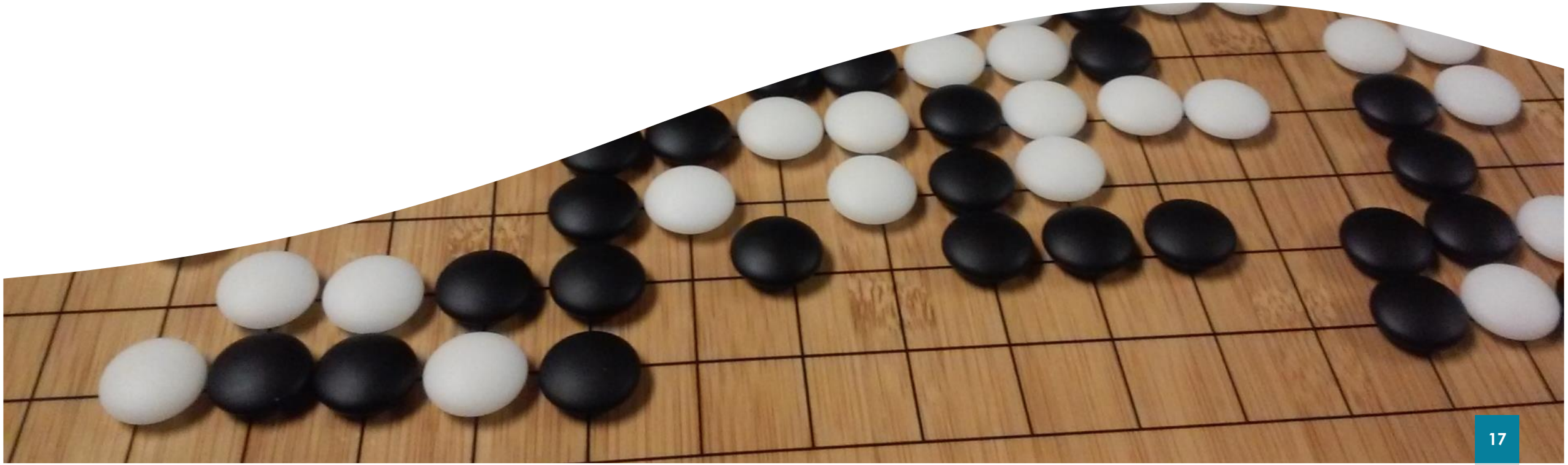
- Charsets and URL encoding
- HTML forms and validation
- HTML Graphics with Canvas and SVG
- HTML 5 Media API - <audio>, <video>
- HTML 5 Geolocation API
- Drag and drop support
- Web Storage API – localStorage and sessionStorage
- Web Workers
- Server Sent Events (SSE)

Quiz Time

https://www.w3schools.com/quiztest/quiztest.asp?qtest=HTML

Exercises

https://www.w3schools.com/html/exercise.asp



Introduction to CSS

Selectors, positioning, backgrounds, box model and layout, grid system, Flexbox

We will follow the CSS Tutorials at W3Schools:

- Selectors
- Positioning
- Colours
- Gradients
- Backgrounds
- Borders
- Box model
- Text and fonts
- Horizontal and vertical alignment

We will follow the CSS Tutorials at W3Schools:

- Complex selectors
- Pseudo-classes and pseudo-elements
- How-to's: navigation bar, dropdowns, gallery
- Forms styling
- Counters
- CSS units
- Selectors specificity
- Web page layout using CSS floating
- Animations and transforms. Styling images

We will follow the CSS Tutorials at W3Schools:

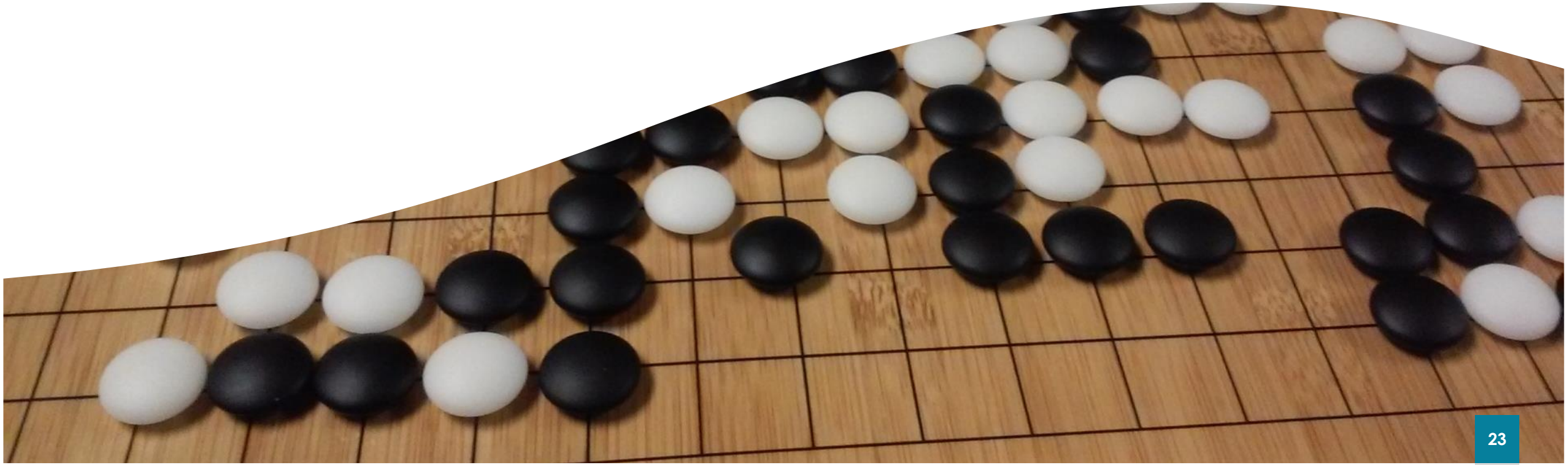
- CSS variables (custom properties) - https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_custom_properties
- calc() Function - https://developer.mozilla.org/en-US/docs/Web/CSS/calc, https://css-tricks.com/creating-color-themes-with-custom-properties-hsl-and-a-little-calc/
- Responsive web design (RWD)
- CSS *@media* rule and media queries
- Flexbox - https://css-tricks.com/snippets/css/a-guide-to-flexbox/
- Responsive page layout using Flexbox
- CSS Grid Layout - https://css-tricks.com/snippets/css/complete-guide-grid/, https://css-tricks.com/auto-sizing-columns-css-grid-auto-fill-vs-auto-fit/

Quiz Time

https://www.w3schools.com/css/css_quiz.asp

Exercises

https://www.w3schools.com/css/exercise.asp



Introduction to SASS

SASS: Syntactically Awesome Style Sheets

We will follow the SASS Tutorial at W3Schools:

- Variables
- Nesting
- @import
- @mixin
- @extend
- Sass functions - string, numeric, list, map, selector, introspection, color
- For more details see: https://sass-lang.com/documentation/style-rules

CSS Modules

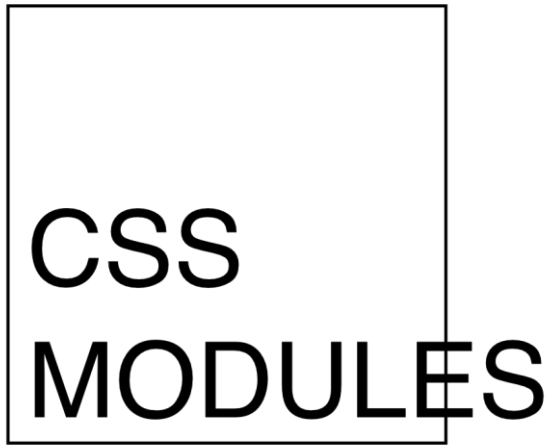
- CSS Modules documentation - https://github.com/css-modules/css-modules
- What are CSS Modules - https://css-tricks.com/css-modules-part-1-need/
- Getting Started with CSS Modules - https://css-tricks.com/css-modules-part-2-getting-started/
- Example:
- /* style.css */

```
.className { color: green; }
```

- /* index.js */

```
import styles from "./style.css"; // import { className } from "./style.css";

element.innerHTML = '<div class="' + styles.className + '">';
```



Homework 1: MyBlogs - Page Layouts

https://github.com/iproduct/fullstack-typescript-react/wiki/Homeworks

Thank's for Your Attention!

Trayan Iliev

IPT – Intellectual Products & Technologies

http://iproduct.org/

http://robolearn.org/

https://github.com/iproduct

https://twitter.com/trayaniliev

https://www.facebook.com/IPT.EACAD