Web Development Review

Roadmap 2017

Web Development 2017

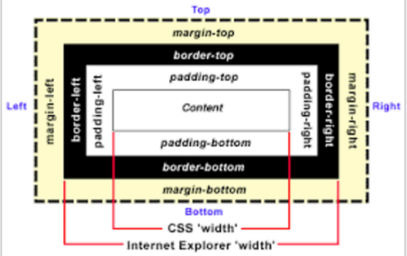
Introducing HTML

We explore the foundations of web and get to grips with the fundamentals of the HTML language. As you will see, its structure and format is relatively straightforward, and you will be able to understand the basics very quickly. We will be focusing on a small number of 'tags' to get started, and also on the ways in which different html files can be linked together to form a site.

Introducing CSS

To introduce 'style' into a page we need another language - Cascading Style Sheets. This is a different language from HTML and is usually stored in a separate file - the stylesheet.

The Box Model

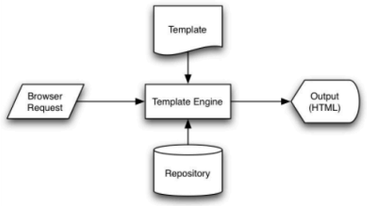


In order to style the same html elements in different ways we need to use classes. This allows us to target specific occurences of an html element for styling purposes. At the heart of the layout engine in web browsers is a concept called the 'box model'. This defines a general layout structure for all HTML elements, providing a language for specifying important dimensions and relationships to other elements.

Navigation

Central to a well design site is a clear and understandably navigation structure. This must easily allow the user to explore the site, provide sufficient context such that the user knows where they are at any stage, and do this in a visually pleasing and efficient manner.

HTML Templates



Review the HTML & CSS constructs covered so far. Introduce html templating using EJS. Refactor a site to use templating techniques.

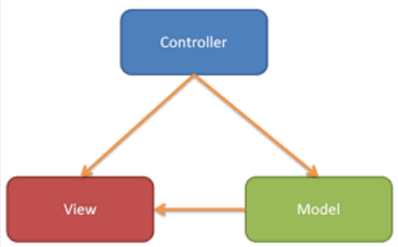
CSS Frameworks

Modern web layouts are not considerably more complex and sophisticated than in the past - particularly as mobile is now considered the 'first' destination for any site. To tackle the complex issues CSS Frameworks have arisen as a convenient way to support multiple browsers and different screen sizes & resolutions.

Starting to Play

To build a web application we need a web framework. This will define the superstructure of our application and provide essential features to enable us to compose a complicated and efficient web application

Model View Controller

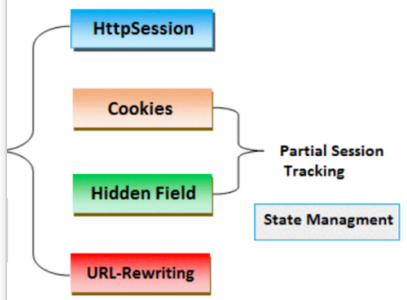


Introducing Models, and exploring how the MVC Triad works

Forms

Providing input to an application is usually encapsulated in a Form. Review the need for forms and explore hwo they are implemented in Play.

7: Sessions

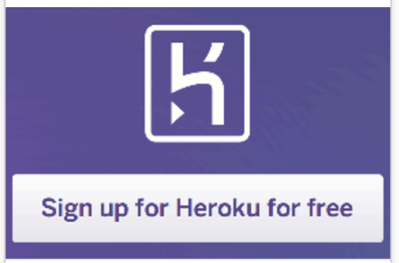


Review the role of Sessions in web applications. Incorporate the hapi-auth-cookie plugin into HAPI application to manage sessions. Review session strategies, protected and unprotected routes, cookie passwords, timeouts and other configuration options.

Git, Github & Bitbucket

Learn the basics of git and github repository management.

Assignment Studio + Deployment



Assignment Studio + Deployment

HTML + CSS Foundations

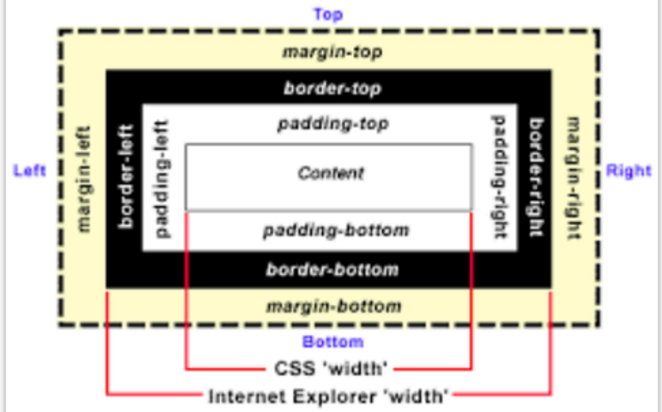
Introducing HTML

We explore the foundations of web and get to grips with the fundamentals of the HTML language. As you will see, its structure and format is relatively straightforward, and you will be able to understand the basics very quickly. We will be focusing on a small number of 'tags' to get started, and also on the ways in which different html files can be linked together to form a site.

Introducing CSS

To introduce 'style' into a page we need another language - Cascading Style Sheets. This is a different language from HTML and is usually stored in a separate file - the stylesheet.

The Box Model



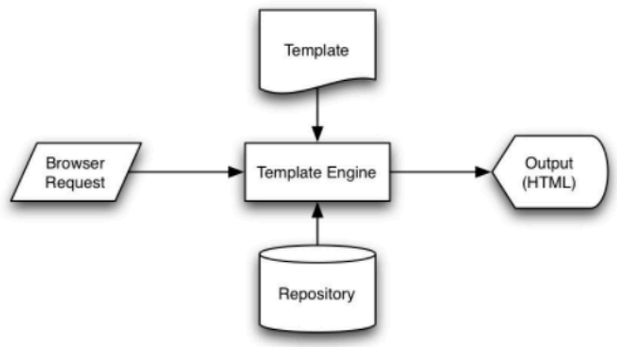
In order to style the same html elements in different ways we need to use classes. This allows us to target specific occurences of an html element for styling purposes. At the heart of the layout engine in web browsers is a concept called the 'box model'. This defines a general layout structure for all HTML elements, providing a language for specifying important dimensions and relationships to other elements.

Navigation, Templates + CSS Frameworks

Navigation

Central to a well design site is a clear and understandably navigation structure. This must easily allow the user to explore the site, provide sufficient context such that the user knows where they are at any stage, and do this in a visually pleasing and efficient manner.

HTML Templates



Review the HTML & CSS constructs covered so far. Introduce html templating using EJS. Refactor a site to use templating techniques.

CSS Frameworks

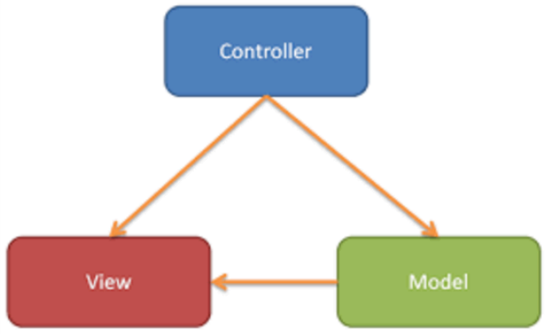
Modern web layouts are not considerably more complex and sophisticated than in the past - particularly as mobile is now considered the 'first' destination for any site. To tackle the complex issues CSS Frameworks have arisen as a convenient way to support multiple browsers and different screen sizes & resolutions.

Web Applications in Play

Starting to Play

To build a web application we need a web framework. This will define the superstructure of our application and provide essential features to enable us to compose a complicated and efficient web application

Model View Controller

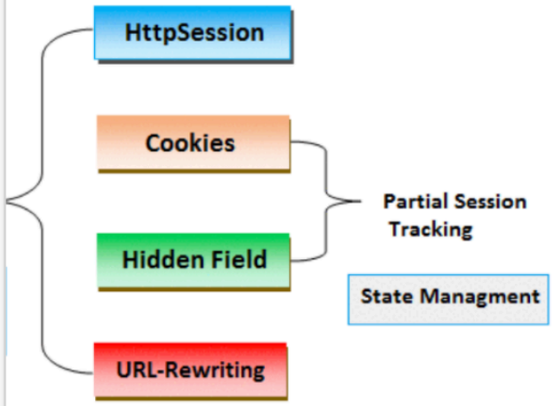


Introducing Models, and exploring how the MVC Triad works

Forms

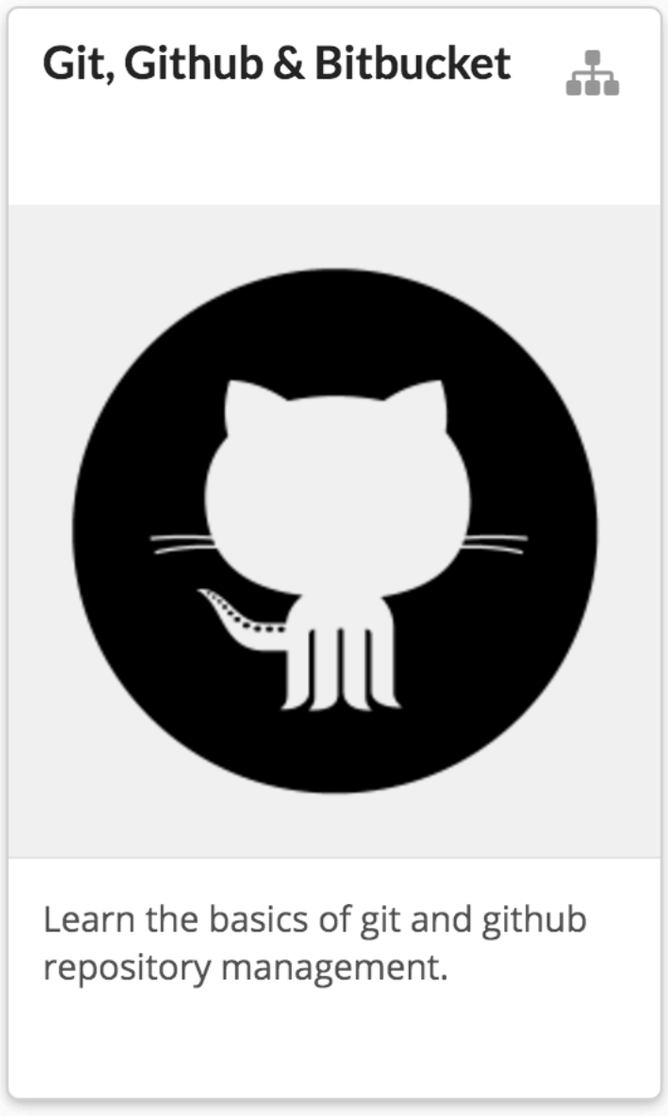
Providing input to an application is usually encapsulated in a Form. Review the need for forms and explore hwo they are implemented in Play.

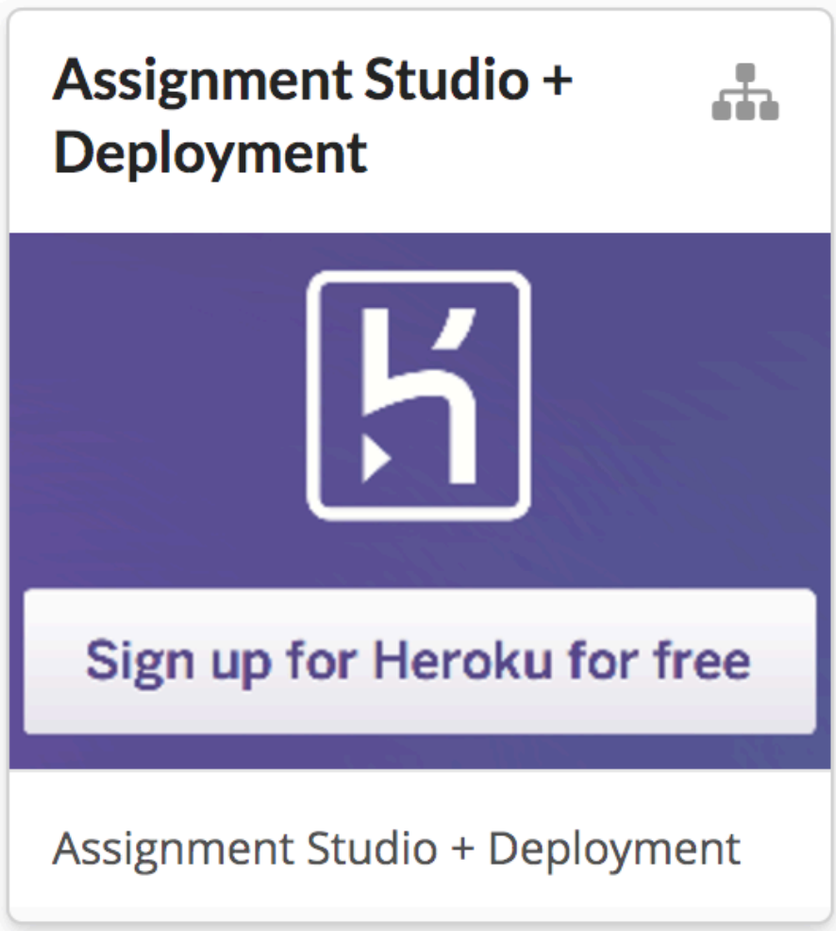
7: Sessions



Review the role of Sessions in web applications. Incorporate the hapi-auth-cookie plugin into HAPI application to manage sessions. Review session strategies, protected and unprotected routes, cookie passwords, timeouts and other configuration options.

Git + Deployment





Introducing HTML

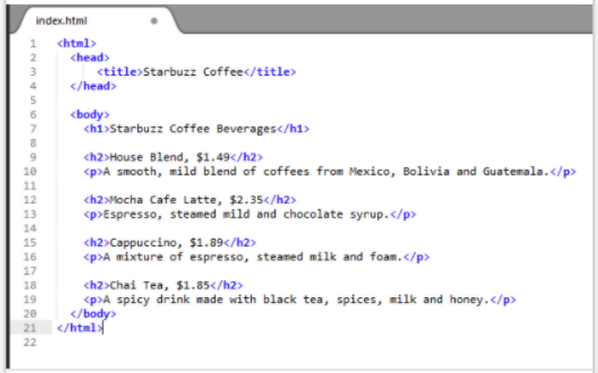
The Nature of the Web

The World Wide Web permeates or lives to an extraordinary degree. However, most of us are unaware of the its technical underpinnings. In this session we explore the nature of the Web and in particular the role of HTTP protocol and the HTML standard. We also identify the 'client/server' nature of the web and its implications. In this context we will gain an understanding of the role of the 'browser' in presenting a web page, and begin to understand pages are retrieved and rendered.

HTML Basics

We can now explore the basics of HTML and we will try to get to grips with the fundamentals of the HTML language. As you will see, its structure and format is relatively straightforward, and you will be able to understand the basics very quickly. We will be focusing on a small number of 'tags' to get started, and also on the ways in which different html files can be linked together to form a site.

Lab-1 Editing HTML


index.html

```
<html>
  <head>
      <title>Starbuzz Coffee</title>
  </head>

  <body>
    <h1>Starbuzz Coffee Beverages</h1>

    <h2>House Blend, $1.49</h2>
    <p>A smooth, mild blend of coffees from Mexico, Bolivia and Guatemala.</p>

    <h2>Mocha Cafe Latte, $2.35</h2>
    <p>Espresso, steamed mild and chocolate syrup.</p>

    <h2>Cappuccino, $1.89</h2>
    <p>A mixture of espresso, steamed milk and foam.</p>

    <h2>Chai Tea, $1.85</h2>
    <p>A spicy drink made with black tea, spices, milk and honey.</p>
  </body>
</html>
```

This lab will firstly introduce you to the tools we will use during the web development module and secondly introduce you to creating, editing, saving and displaying a web page.

Lab-1 HTML Structure

The labs are where you will do the real learning in this module. In Lab0-01 you will become familiar with the editor Sublime. We will then use this editor to create a simple multi-page web site containing a small variety of text, images and links. In this lab we will explore some of the basic features of CSS, including colours, font and interesting techniques for setting styles across entire sections of a page.

Introducing CSS

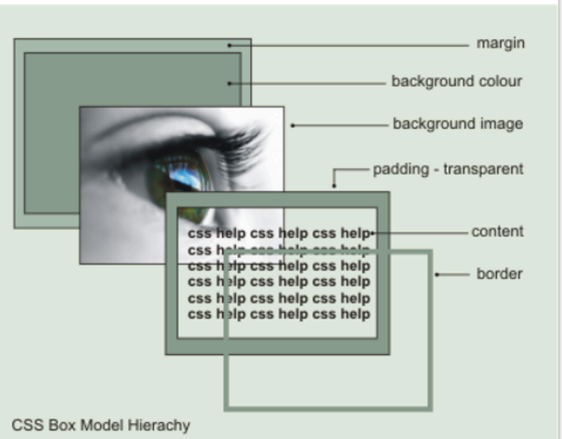
HTML Elements

HTML Elements are at the heart of the HTML specification. There are perhaps a dozen categories or elements to explore -in Lab-01 we explored elements from 5 of these categories. Here we review these and examine the role and format of each one in detail.

CSS Basics

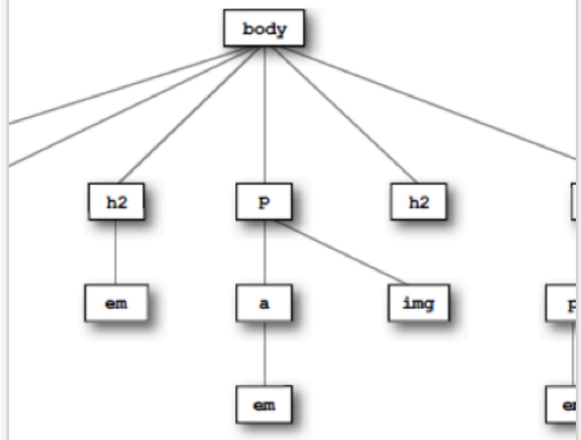
This language looks simple at first, but as we will see in the next few weeks, it is considerably more complex than HTML and will require a very careful approach to get right

CSS Rules



In order to style the same html elements in different ways we need to use classes. This allows us to target specific occurences of an html element for styling purposes.

CSS Cascade



Inheritance is a key feature in how CSS applies rules, and it has some interesting side effects

Lab-2 CSS Intro

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>APP Store</title>
    <link type="text/css" rel="stylesheet" href="style.css" media="screen" />
  </head>
  <body>
    <img src="images/score.png"/><h1>Score: Apps, Movies, Music, Books</h1>
    <ol>
      <li><a href="apps.html">Apps</a></li>
      <li><a href="music.html">Music</a></li>
      <li><a href="movies.html">Movies</a></li>
    </ol>
    <div class="main_panel">
      <h2>New Games</h2>
        <ul>
          <li><img src="images/delete.png"/>Clear All</li>
          <li><img src="images/google.jpg"/>Google Box</li>
          <li><img src="images/squink.jpg"/>Squinks</li>
        </ul>
    </div>
  </body>
</html>
```

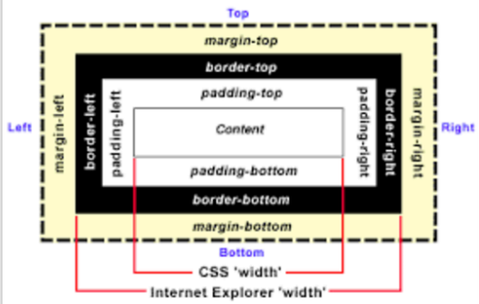
In this lab we will begin a new project which will have CSS stylesheet from the beginning. We will explore some of the basic features of CSS, including colours, font and interesting techniques for setting styles across entire sections of a page.

CSS Box Model

Classes, IDs & Divs

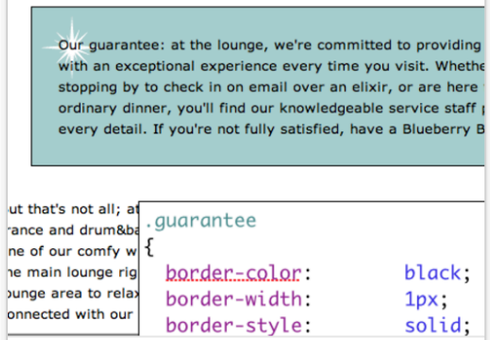
These three concepts are essential in order to build useful style sheets. The allow is to define different regions, elements a subsets of elements on a page.

Box Fundamentals



At the heart of the layout engine in web browsers is a concept called the 'box model'. This defines a general layout structure for all HTML elements, providing a language for specifying important dimensions and relationships to other elements.

Box Model Example



A worked example of the box model in action. This example featured in this weeks lab, and demonstrates the major features of the model.

Multicolumn Layout

Using an understanding of the fundamental features of the box model we can start to produce more interesting page layouts. This will allow is to grow multi-column pages that can vary according to the size of the browser windows used to view them.

Project 1 Specification

When you finish these labs you will have the knowledge to prepare your first project for this module. The briefing is here - pay close attention to the guidelines

Lab-3a Layout

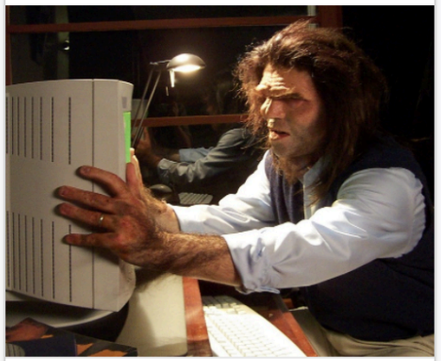
In this lab we will work towards developing a site with a more ambitious use of CSS. In particular, tbe box model will be used to achieve a more professional user experience.

Lab-3b Multicolumn

This weeks lab will give you more practical experience od the box model and specifically how to build a simple multi-column layout using the techniques we have explored in class. You should complete this lab before starting to consider your project in detail.

Navigation

The Evolution of the Web



Both HTML & CSS continue to evolve. Understanding where it is going is part and parcel of becoming a programmer. Here we look a some aspects of the history of HTML, including some major milestones in its evolution.

Navigation

Central to a well design site is a clear and understandably navigation structure. This must easily allow the user to explore the site, provide sufficient context such that the user knows where they are at any stage, and do this in a visually pleasing and efficient manner.

Semantic HTML

HTML5, the latest version of the standard, introduced a range of new elements. Among the most interesting are the so-called 'semantic' elements. These attempt to re-examine the proliferation of DIVs in html, and proposed an alternative vocabulary that would better reflect the purpose of many of these DIVs

HTML/CSS Style Guide

```
<ul>
  <li>Fantastic
  <li>Great
</ul>
```

```
.example {
  color: blue;
}
```

A review of some of the guidelines from the Google HTML/CSS style recommendations.

Lab-4a Navigation

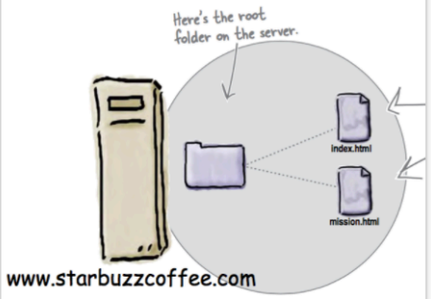
Restructure the output of the last lab to participate in a tabbed navigation structure, and then produce a simple application mockup using these techniques. Do this by introducing a simple tabbed design into the site to provide the user with the visual metaphor for navigation. We then 'wire-up' these tabs to lead the user through the site. We will do this twice: once for the example content we laid out in last weeks lab. Then we will apply this to the app store site, using the same CSS rules.

Lab-4b Case Study

Explore a web site that embodies many of the techniques we have explored so far. In the lab the web is evolved from unstyled content to a reasonably elegant and clean design - using semantic html element where appropriate.

HTML Templating

Deployment

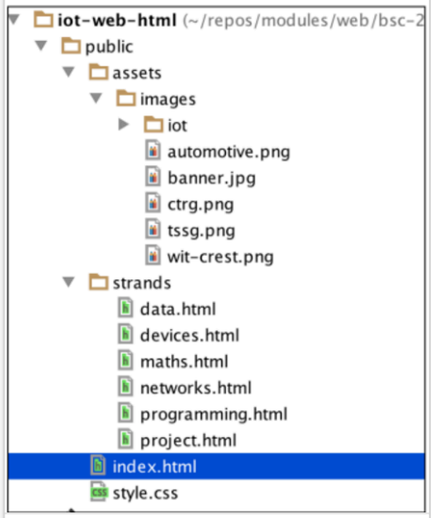


Deploying a site involves understanding a little more about Clients & Servers, Hosting Providers, Domain Names, Transferring the Sites Files, HTTP and Absolute & Relative Paths

Harp & Surge

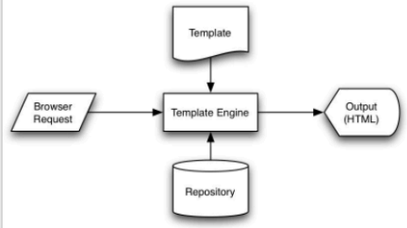
Harp.js and Surge.sh are the two services we will use to server the page locally, and also to deploy it to a public web server.

Case Study



A review of the the evolution of a web site from simple unstyled pages to a reasonable simple but appealing layout.

Templates



The EJS template system implements mechanisms for assembling sites from templates - which are called 'partials' in EJS. Additionally, there is a complimentary 'layout' mechanism for reusing entire page structures.

Lab-5a Harp & Surge

Install software tools to serve a web site locally and also to deploy the web site to a public web server.

Lab-5b Templating

```
├── harp.json
├── public
│   ├── assets
│   │   ...
│   ├── includes
│   │   ├── _curriculum.ejs
│   │   ├── _footer.ejs
│   │   ├── _header.ejs
│   │   ├── _sponsors.ejs
│   │   └── _summary.ejs
│   ├── index.ejs
│   ├── strands
│   │   ├── _layout.ejs
│   │   ├── data.ejs
│   │   ├── devices.ejs
│   │   ├── maths.ejs
│   │   ├── networks.ejs
│   │   ├── programming.ejs
│   │   └── project.ejs
│   └── style.css
```

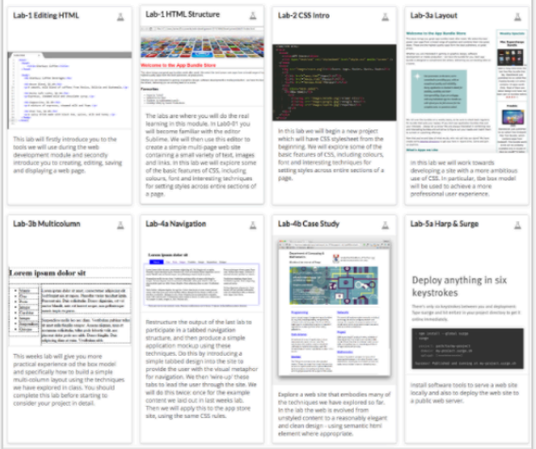
Rebuild the IoT web site from thee last lab using templating. This version of the site will aim to significantly reduce the content the author has to manage by reusing 'templates' containing common sections.

Lab-5c Navigation

Rework the tabbed navigation site from lab 04 to use EJS template

CSS Frameworks

Lab Review



Review of the Labs to date

Semantic-UI I

	Mobile	Tablet
Width	100%	723px
Gutter Size	1em	Fluid
Responsive Visibility	`mobile only`	`tablet only`
Device Width	below 768px	768px - 991px

An overview of the container and segments styles in the framework

Semantic-UI II

An exploration of the Grid and Image styles.

Semantic-UI III

Employee	Correct Guesses
Lena Human Resources	22
Matthew Fabric Design	15
Lindsay Entertainment	12
Mark Executive	11

An exploration of the Tables

Semantic Part IV

Our final tour of the Semantic-UI library - with a focus on icons, variations in segments and responsive grids

Lab-6a Semantic UI

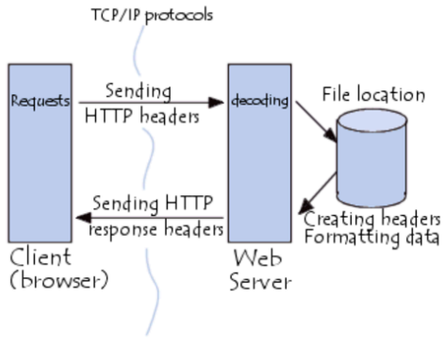
Rebuild the iot web site using semantic-ui

Lab-6b Semantic UI

Continue to enhance the IoT web site with additional Semantic-UI styles & Components

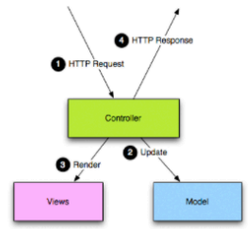
Starting to Play

Dynamic Web Applications



The applications to date have been static - essentially a collecting of individual web pages. For more sophisticated web site we need to move to creating web applications. This will require a shift to considering the web site as an aggregating of fragments of web pages, composed and assembled by a program we also write.

Starting to Play



To build a web application we need a web framework. This will define the superstructure of our application and provide essential features to enable us to compose a complicated and efficient web application

Web App Introduction

Structure of a web app: Front-end Vs Backend. Routers, Models, Views, Controllers

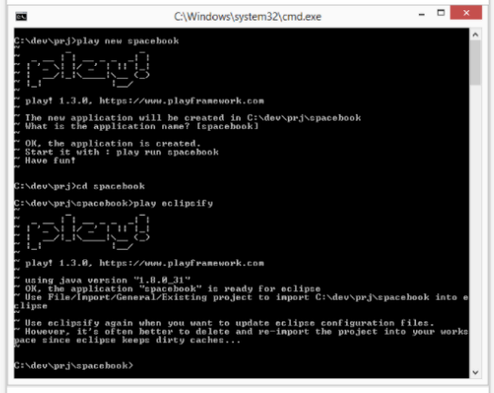
Front-end

Review the front end of a Play Project, consisting of view, layouts and partials.

Back-end

The back end of a Play project consists of routes, controllers, models + configuration.

Lab-07a Play Setup



Review the spacebook-semantic project from an earlier lab. Become familiar with the Command Prompt. Install 2 new programming environments: a Web Framework called Play and an Integrated Development Environment called Eclipse.

Lab-07b Playlist 1

Import, renams and run a new starter project. Extend the project to include a Model. Rework the views to display the model.

Model View Controller

Views

Beethoven Sonatas

Song	Artist
Piano Sonata No. 3	Beethoven
Piano Sonata No. 7	Beethoven
Piano Sonata No. 10	Beethoven

Beethoven Concertos

Song	Artist
Piano Concerto No. 1	Beethoven
Piano Concerto No. 12	Beethoven
Piano Concerto No. 23	Beethoven

Explore the play templating language in more depth

Models

SELECT * FROM SONG;

ID	ARTIST	DURATION	TITLE
1	Beethoven	0	Piano Sonata No. 3
2	Beethoven	0	Piano Sonata No. 7
3	Beethoven	0	Piano Sonata No. 10
4	Beethoven	0	Piano Concerto No. 27
5	Beethoven	0	Piano Concertos No. 17
6	Beethoven	0	Piano Concerto No. 10

(6 rows, 6 ms)

Edit

Playu includes a simple relational database - we can preload and read/write from this db in our app.

Lab-08a Playlist-2

Move the playlist model into the database. Prime the database from a YAML file. Modify the Model classes to work with this database. Render the models from the database.

Lab-08b Playlist-3

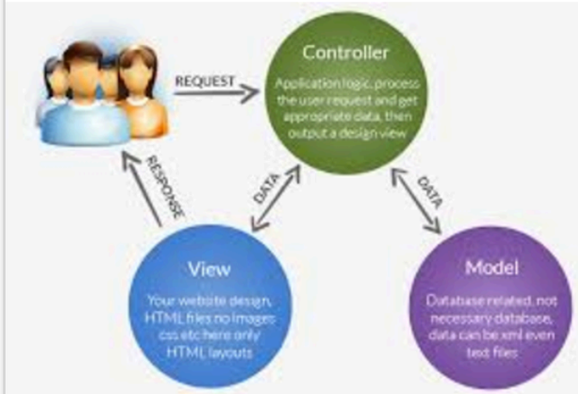
Move the playlists into their own view. Introduce routes for opening the playlist and also deleting individual songs.

Forms

Todo

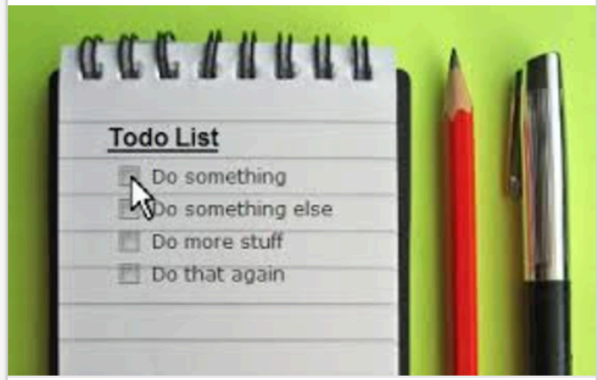
A Separate case study - a simple todo list application

Playlist Features



Complete a review of the playlist features

Lab-09b Todolist-1



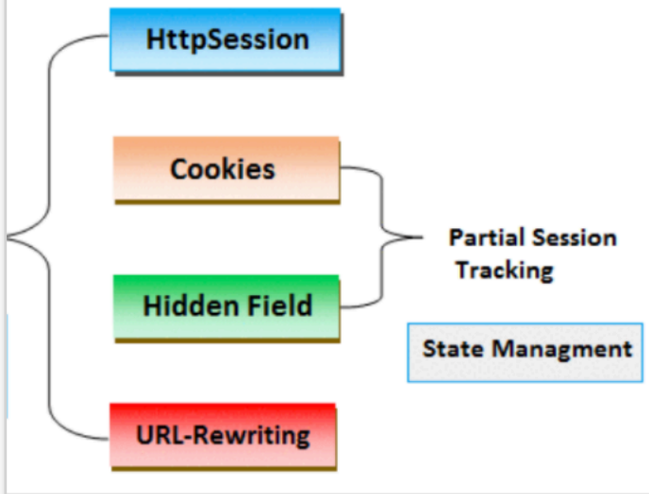
Develop a completely new application, using the techniques we have explored so far.

Lab-09a Playlist-4

Introduce forms into a play application to enable the user to create playlists.

Sessions

Sessions

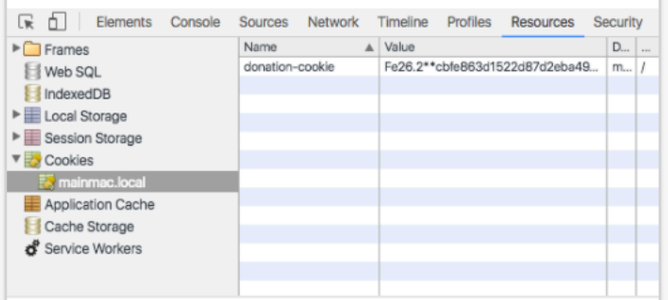


Keeping track of the currently logged in user is a challenge - as HTTP is, by definition 'stateless'. Hidden form fields, url rewriting and cookies are three common techniques for implementing sessions.

Sessions in Todo

Include the session on the Todo app

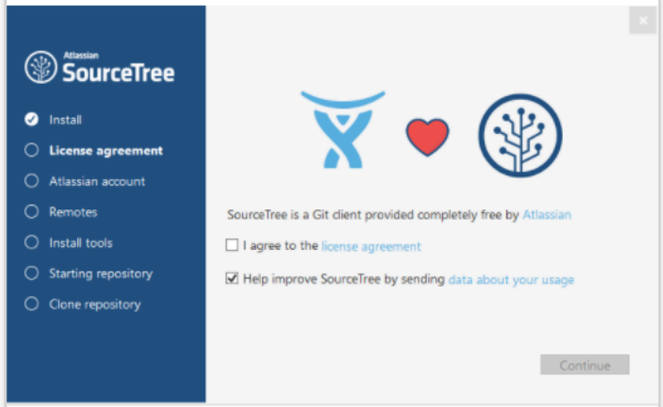
Lab-10 Todolist-2



Incorporate sessions tracking into the todo app

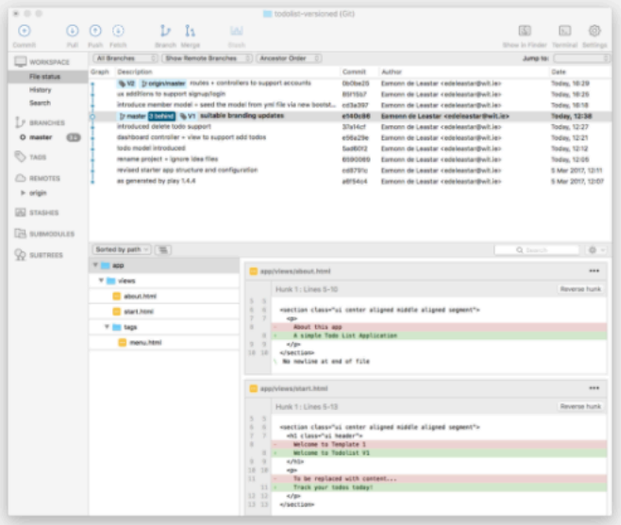
Git

Lab-11a Sourcdtree



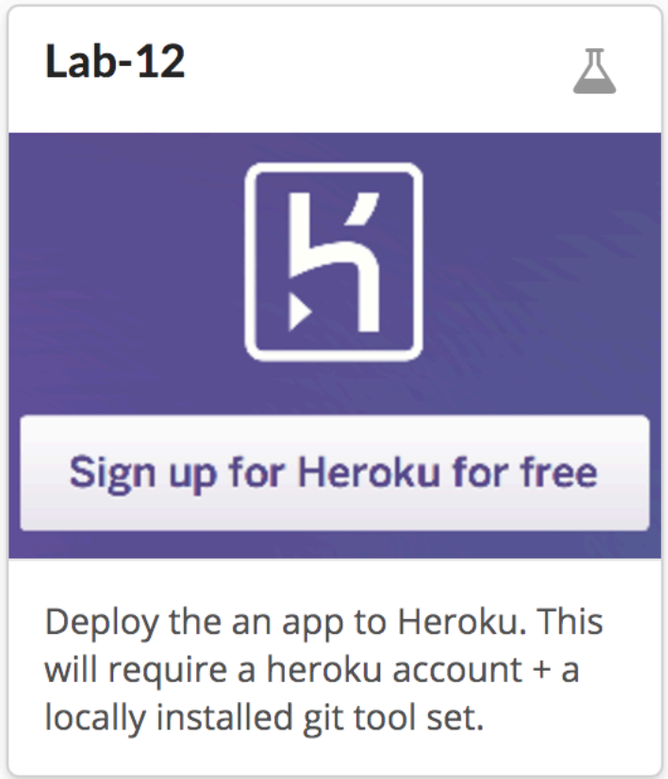
Install and Configure Sourcetree

Lab-11b Todolist-3



Rebuild Todolist - this time committing to git version control as the app is composed.

Deployment

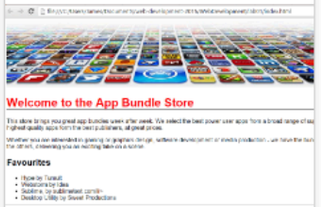

Lab-12
Sign up for Heroku for free
Deploy the an app to Heroku. This will require a heroku account + a locally installed git tool set.

Front-end Labs

Lab-1 Editing HTML

This lab will firstly introduce you to the tools we will use during the web development module and secondly introduce you to creating, editing, saving and displaying a web page.

Lab-1 HTML Structure



The labs are where you will do the real learning in this module. In Lab0-01 you will become familiar with the editor Sublime. We will then use this editor to create a simple multi-page web site containing a small variety of text, images and links. In this lab we will explore some of the basic features of CSS, including colours, font and interesting techniques for setting styles across entire sections of a page.

Lab-2 CSS Intro

In this lab we will begin a new project which will have CSS stylesheet from the beginning. We will explore some of the basic features of CSS, including colours, font and interesting techniques for setting styles across entire sections of a page.

Lab-3a Layout

In this lab we will work towards developing a site with a more ambitious use of CSS. In particular, tbe box model will be used to achieve a more professional user experience.

Lab-3b Multicolumn

This weeks lab will give you more practical experience od the box model and specifically how to build a simple multi-column layout using the techniques we have explored in class. You should complete this lab before starting to consider your project in detail.

Lab-4a Navigation

Restructure the output of the last lab to participate in a tabbed navigation structure, and then produce a simple application mockup using these techniques. Do this by introducing a simple tabbed design into the site to provide the user with the visual metaphor for navigation. We then 'wire-up' these tabs to lead the user through the site. We will do this twice: once for the example content we laid out in last weeks lab. Then we will apply this to the app store site, using the same CSS rules.

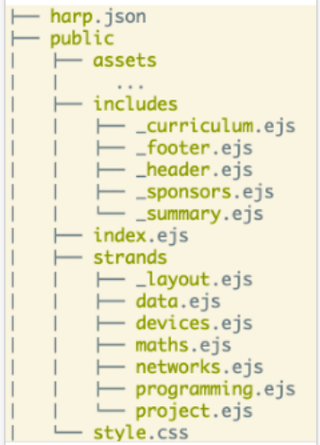
Lab-4b Case Study

Explore a web site that embodies many of the techniques we have explored so far. In the lab the web is evolved from unstyled content to a reasonably elegant and clean design - using semantic html element where appropriate.

Lab-5a Harp & Surge

Install software tools to serve a web site locally and also to deploy the web site to a public web server.

Lab-5b Templating



Rebuild the IoT web site from thee last lab using templating. This version of the site will aim to significantly reduce the content the author has to manage by reusing 'templates' containing common sections.

Lab-5c Navigation

Rework the tabbed navigation site from lab 04 to use EJS template

Lab-6a Semantic UI

Rebuild the iot web site using semantic-ui

Lab-6b Semantic UI

Continue to enhance the IoT web site with additional Semantic-UI styles & Components

Back-end Labs

Lab-07a Play Setup

Review the spacebook-semantic project from an earlier lab. Become familiar with the Command Prompt. Install 2 new programming environments: a Web Framework called Play and an Integrated Development Environment called Eclipse.

Lab-07b Playlist 1

Import, renams and run a new starter project. Extend the project to include a Model. Rework the views to display the model.

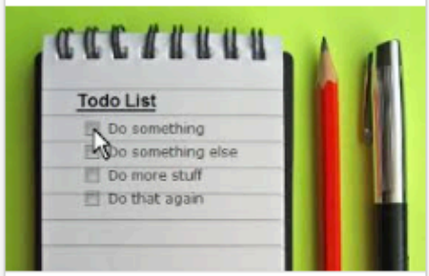
Lab-08a Playlist-2

Move the playlist model into the database. Prime the database from a YAML file. Modify the Model classes to work with this database. Render the models from the database.

Lab-08b Playlist-3

Move the playlists into their own view. Introduce routes for opening the playlist and also deleting individual songs.

Lab-09b Todolist-1

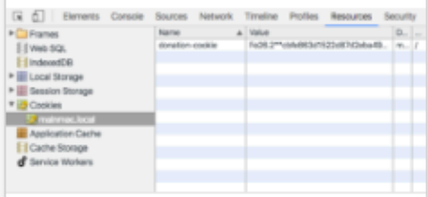


Develop a completely new application, using the techniques we have explored so far.

Lab-09a Playlist-4

Introduce forms into a play application to enable the user to create playlists.

Lab-10 Todolist-2

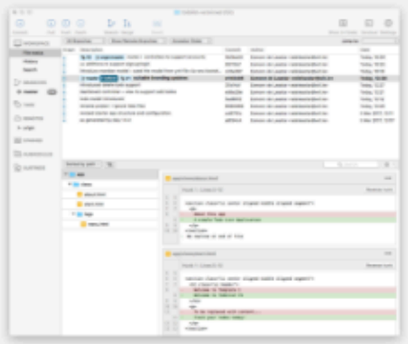


Incorporate sessions tracking into the todo app

Lab-11a Sourcdtree

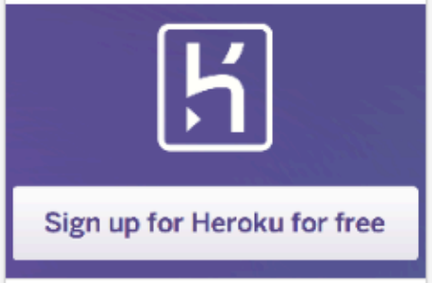
Install and Configure Sourcetree

Lab-11b Todolist-3



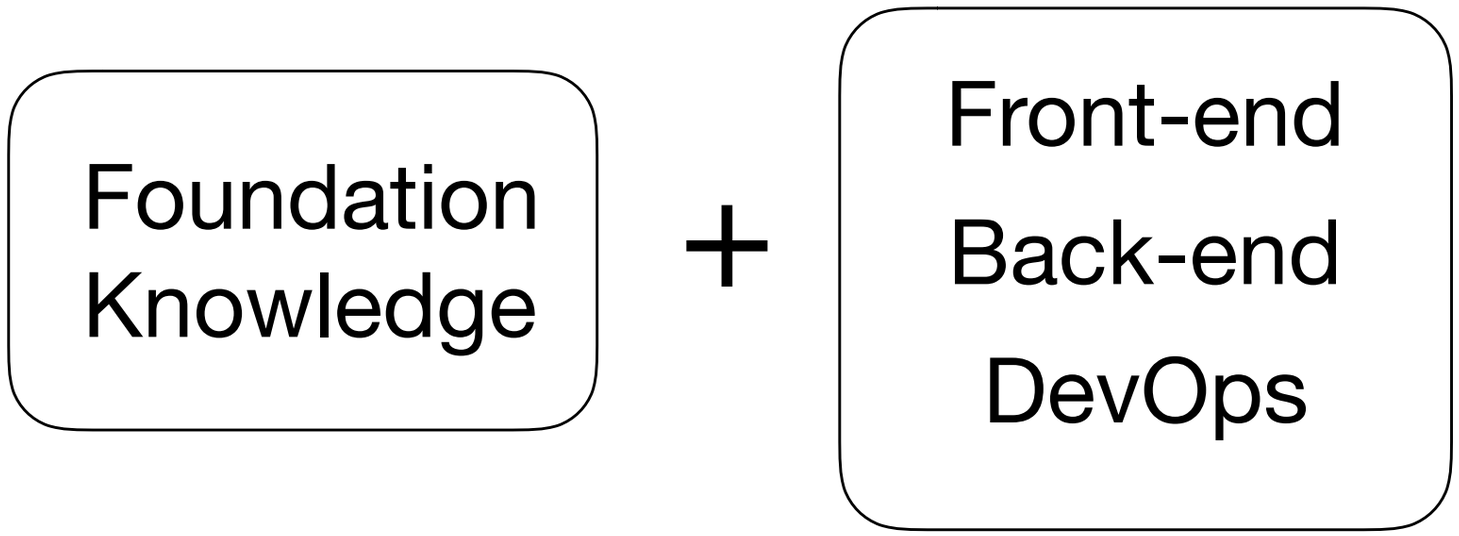
Rebuild Todolist - this time committing to git version control as the app is composed.

Lab-12



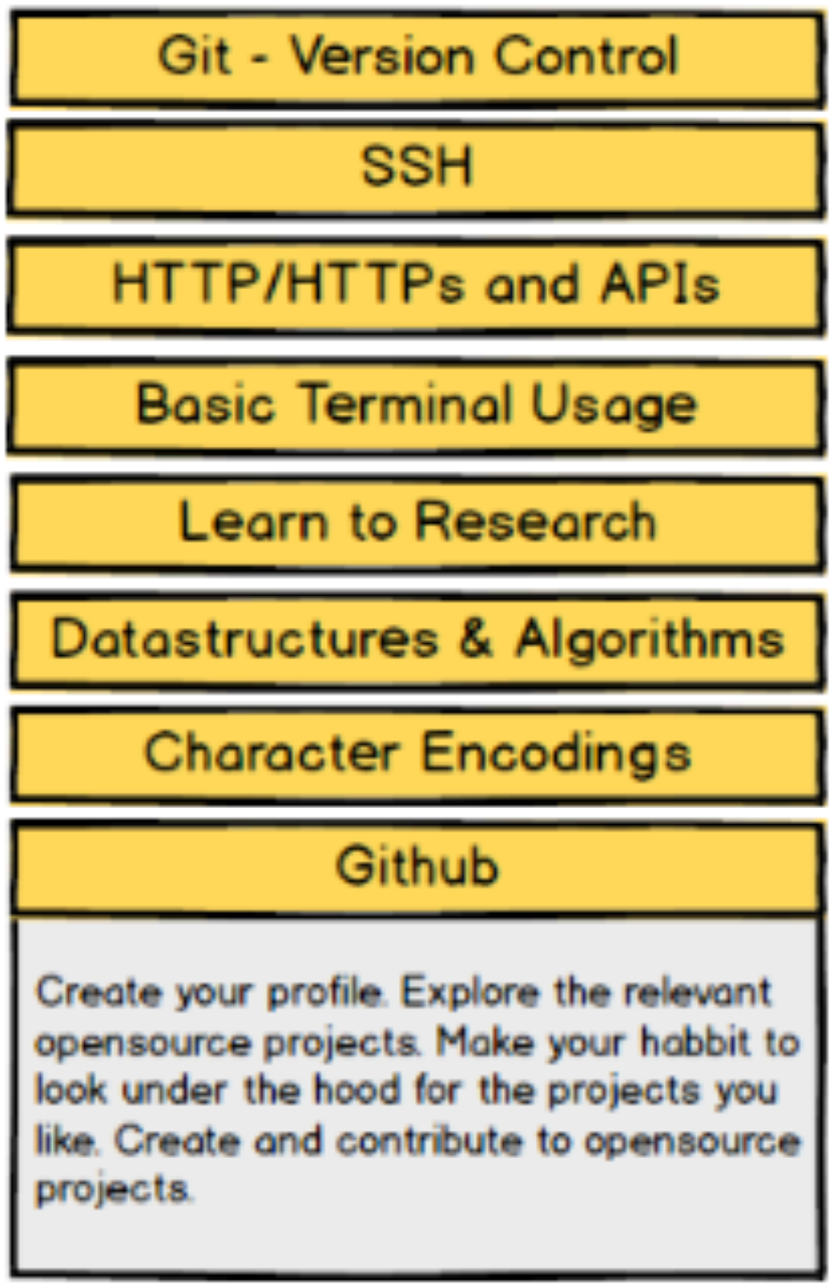
Deploy the an app to Heroku. This will require a heroku account + a locally installed git tool set.

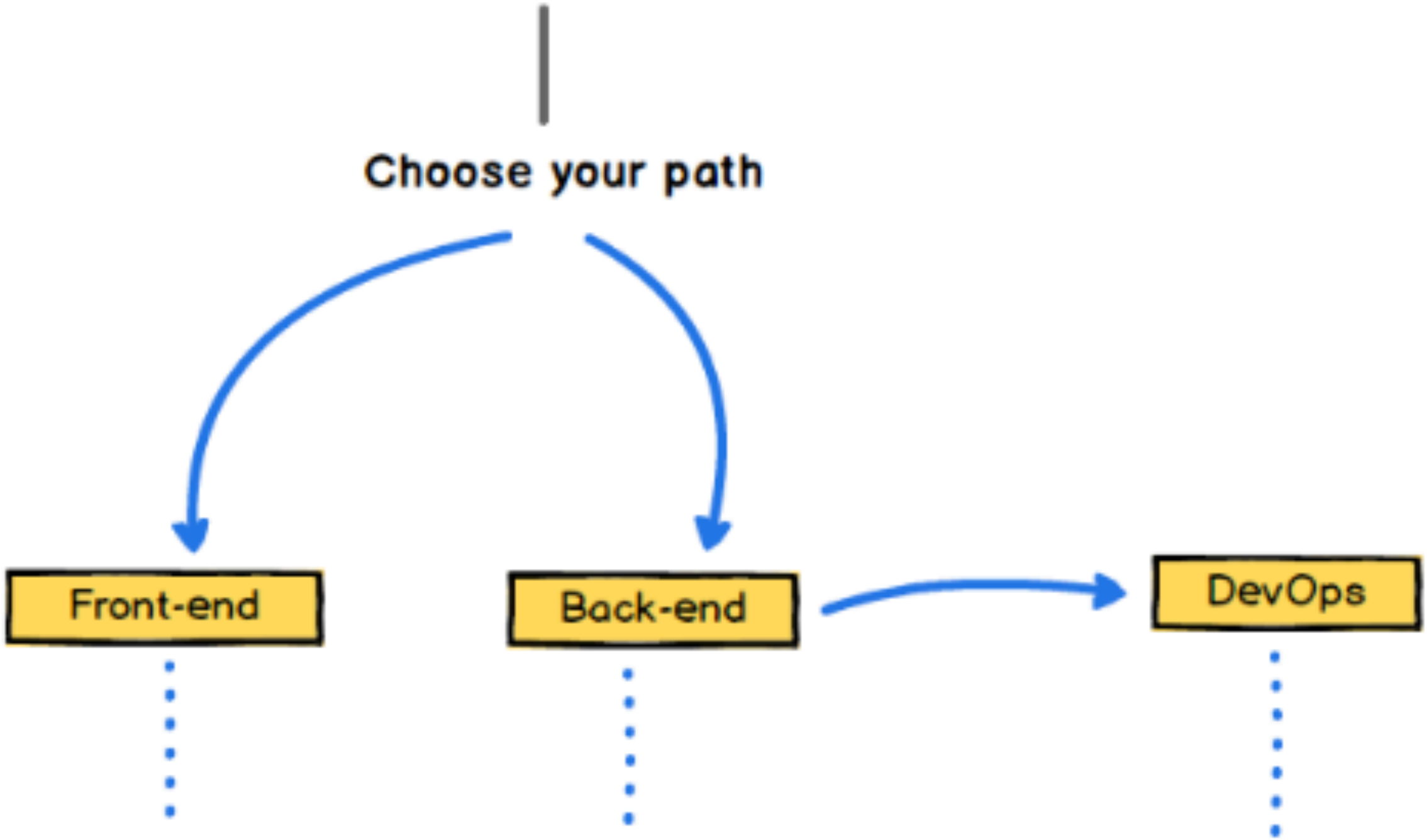
A roadmap to becoming a web developer in 2017



https://github.com/kamranahmedse/developer-roadmap

Foundation Knowledge





Front-end + Back-end + DevOps == Full Stack

Front-end

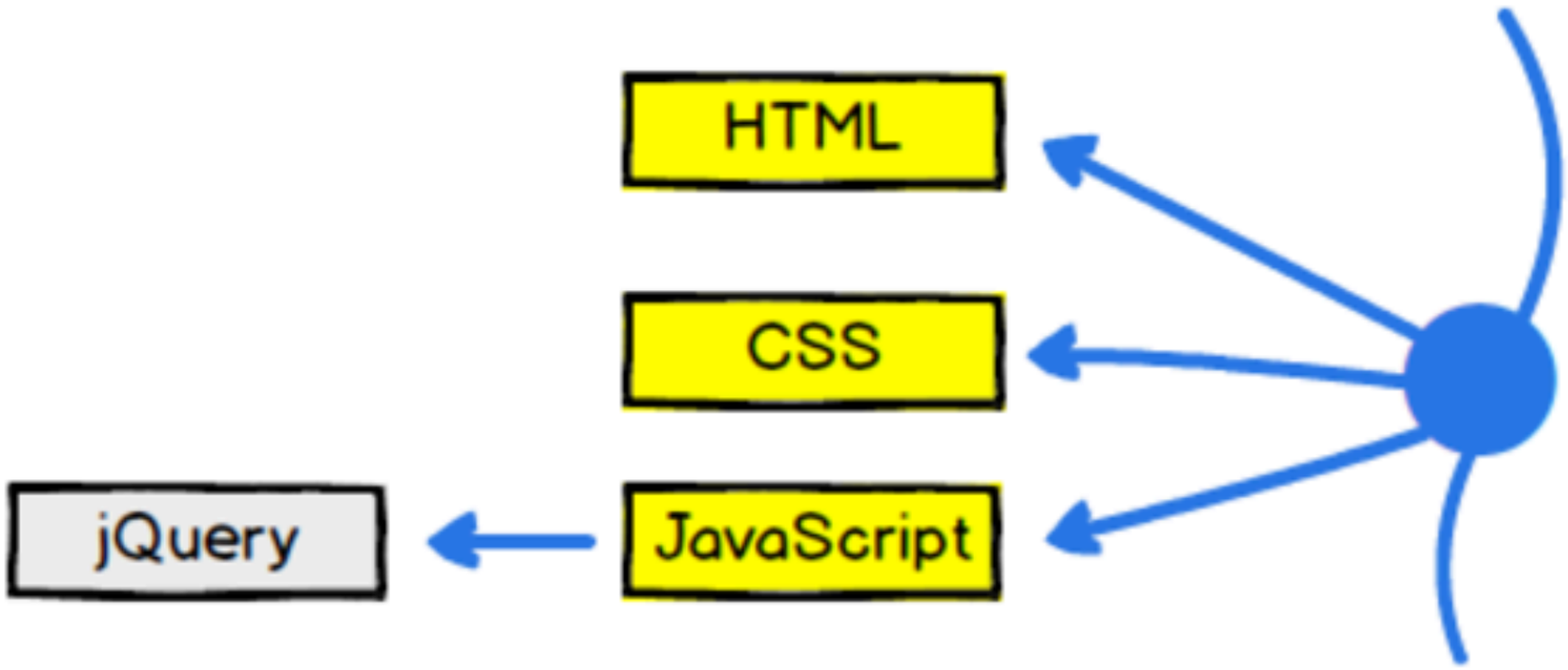
Basics

CSS Frameworks & Processors

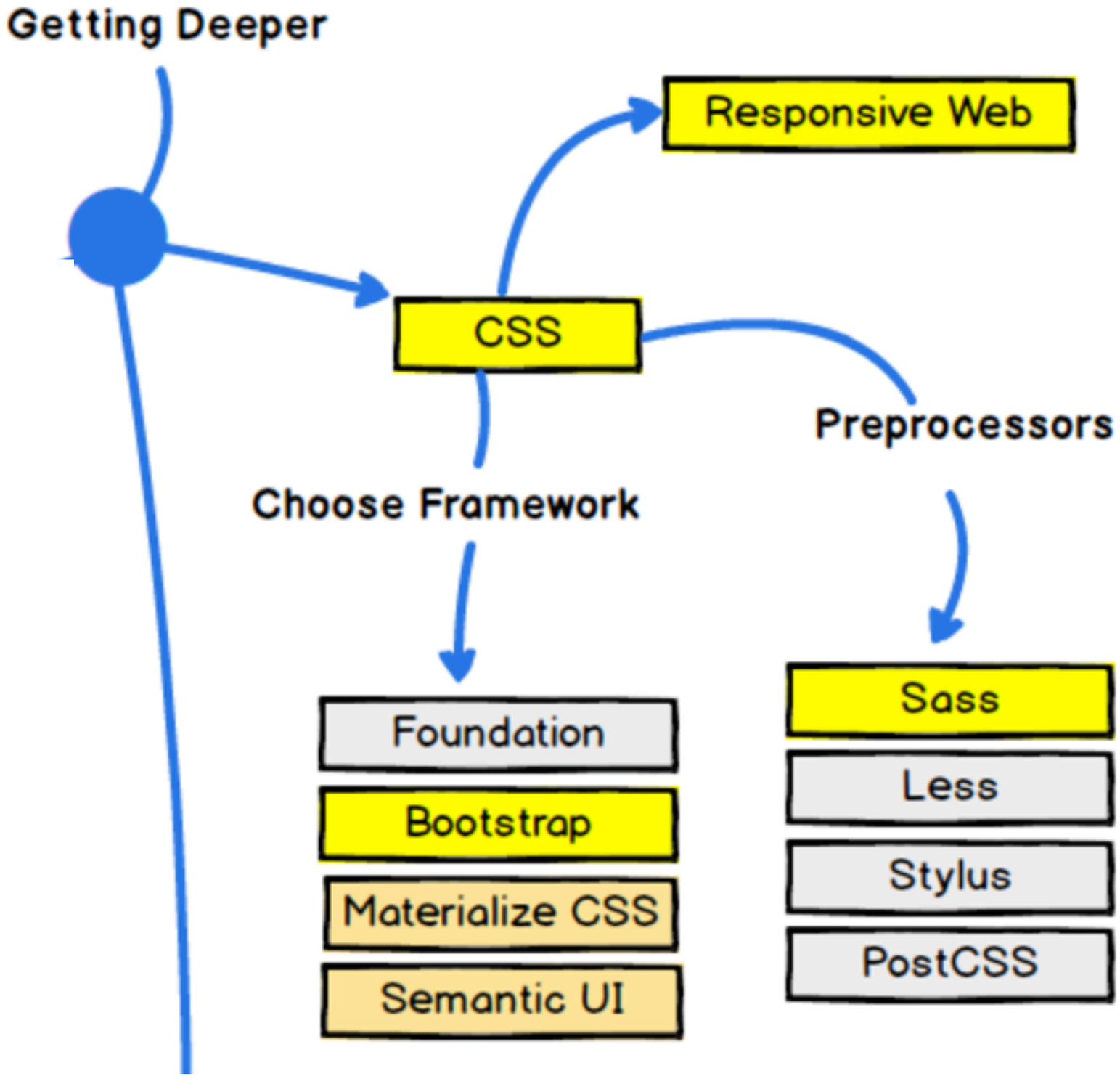
JS Client Frameworks + Tools

CSS Mastery, Graphics & Visualisation

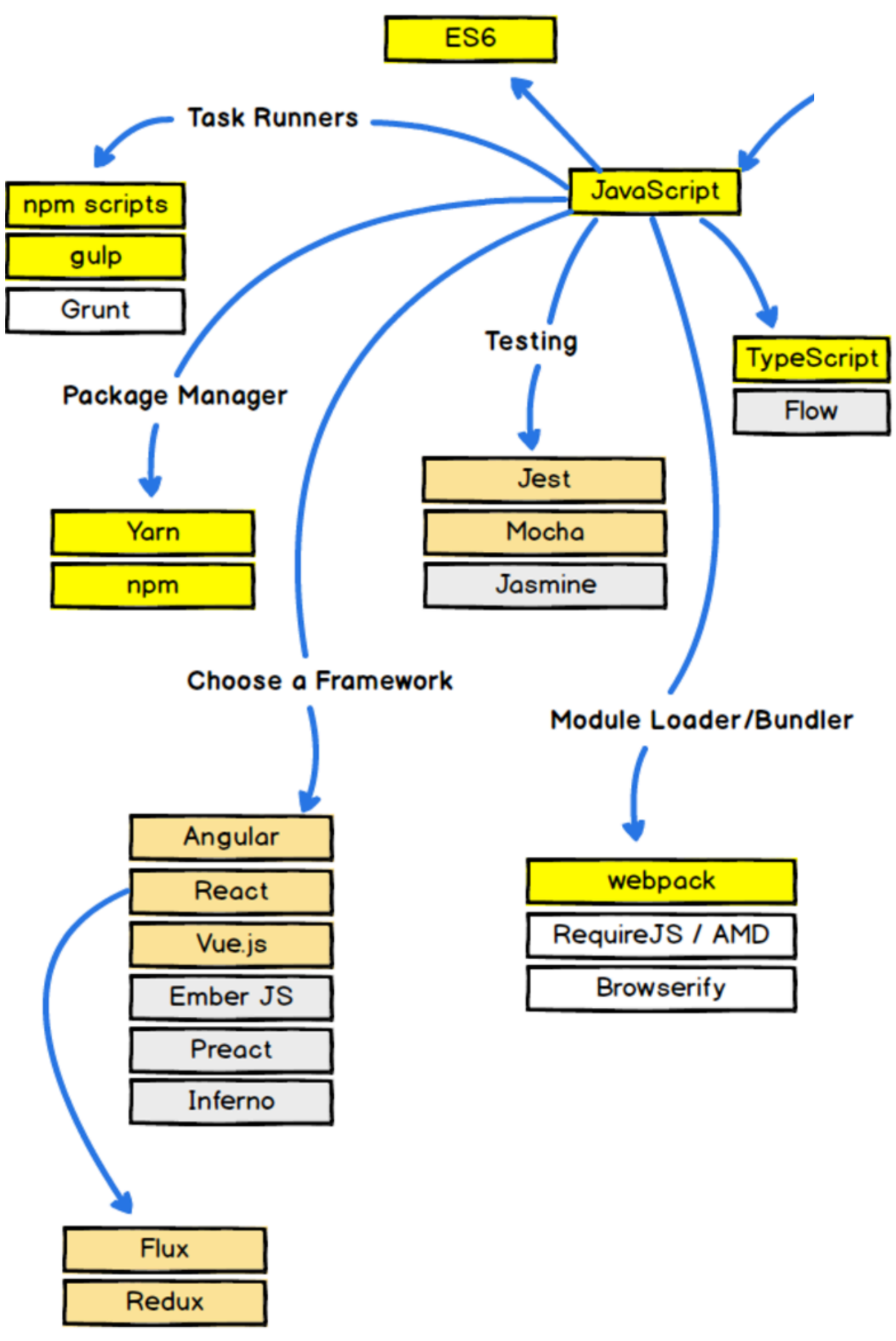
Front-end: Basics



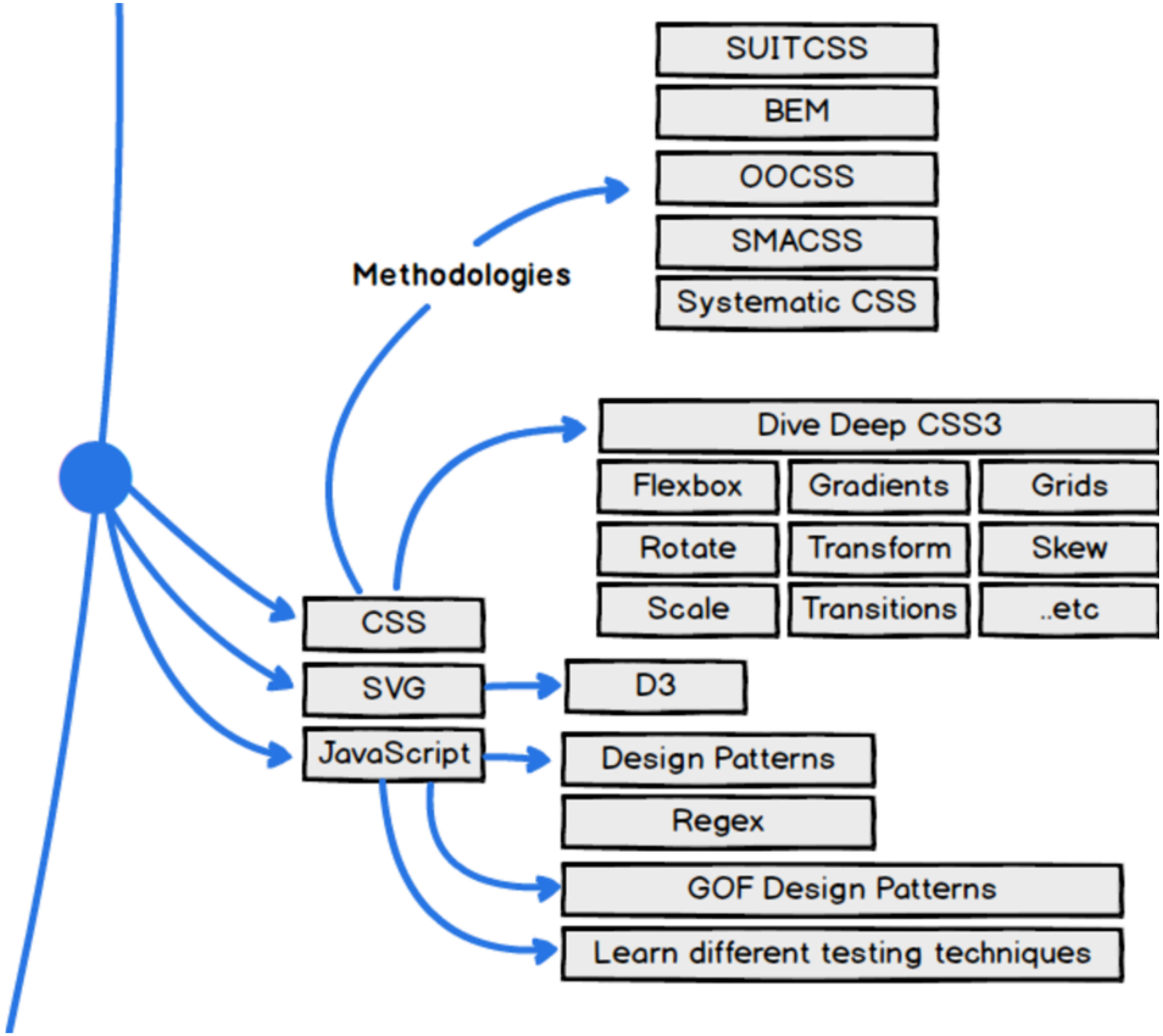
Front-end: CSS Frameworks & Processors



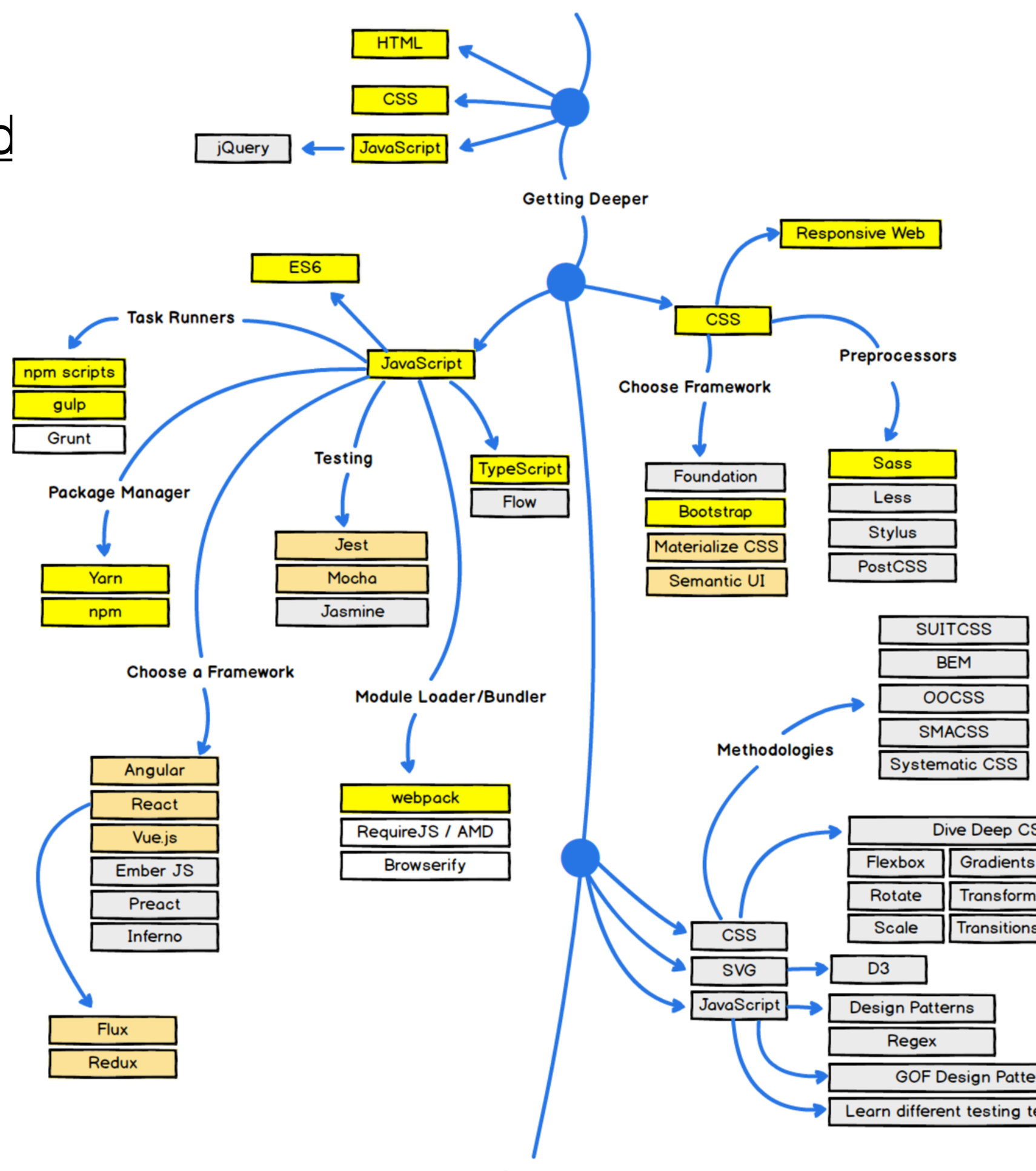
Front-end: JS Client Frameworks, & Tools



Front-end: CSS Mastery, Graphics & Visualisation



Front-end


HTML
CSS
JavaScript
jQuery
Getting Deeper
ES6
Task Runners
npm scripts
gulp
Grunt
JavaScript
Package Manager
Yarn
npm
Testing
Jest
Mocha
Jasmine
TypeScript
Flow
Choose a Framework
Angular
React
Vue.js
Ember JS
Preact
Inferno
Flux
Redux
Module Loader/Bundler
webpack
RequireJS / AMD
Browserify
CSS
Responsive Web
Preprocessors
Choose Framework
Foundation
Bootstrap
Materialize CSS
Semantic UI
Sass
Less
Stylus
PostCSS
Methodologies
SUITCSS
BEM
OOCSS
SMACSS
Systematic CSS
CSS
Dive Deep CSS3
Flexbox
Gradients
Grids
Rotate
Transform
Skew
Scale
Transitions
..etc
SVG
D3
JavaScript
Design Patterns
Regex
GOF Design Patterns
Learn different testing techniques

Back-end

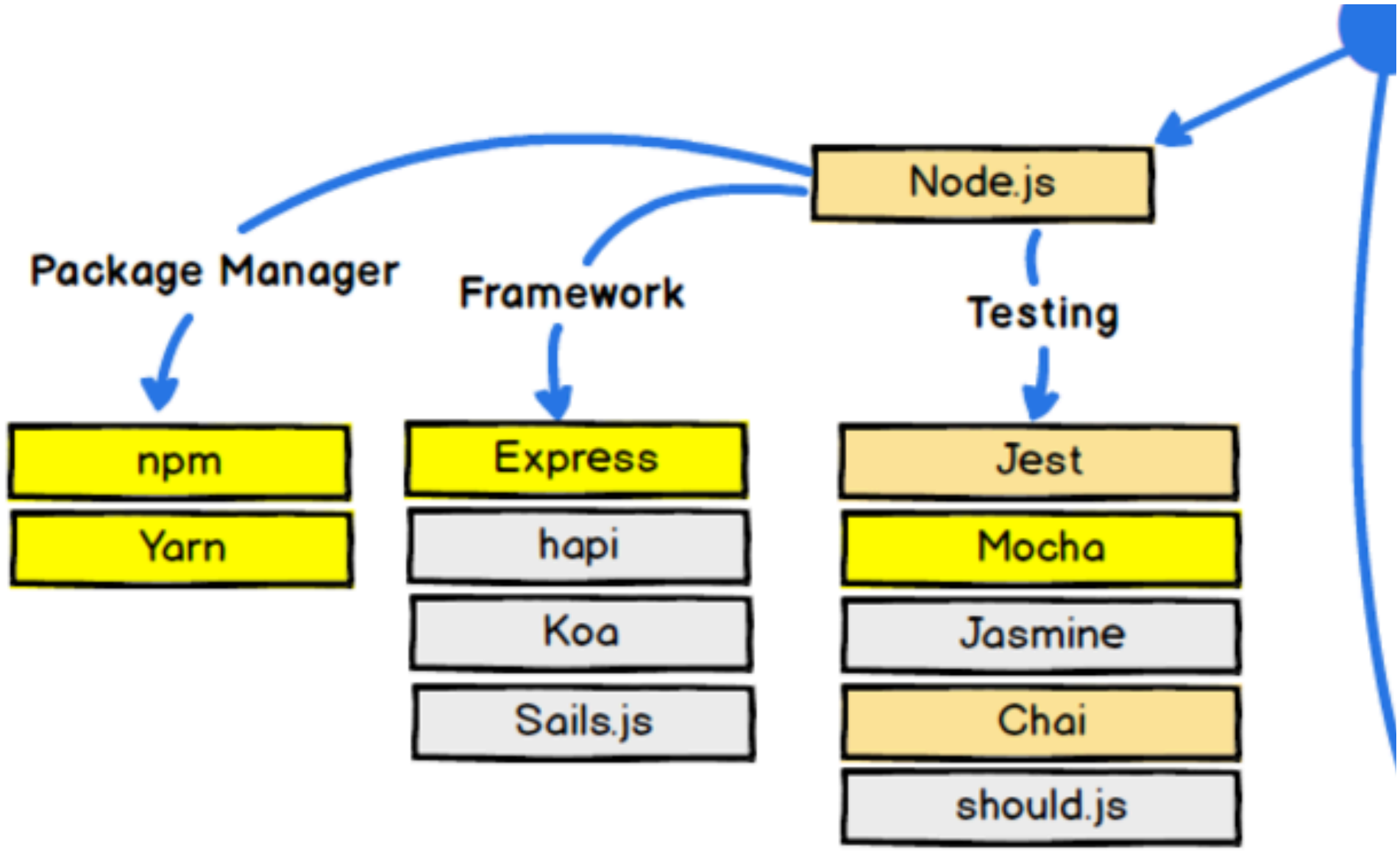
Node.js

Infrastructure + Key Techniques

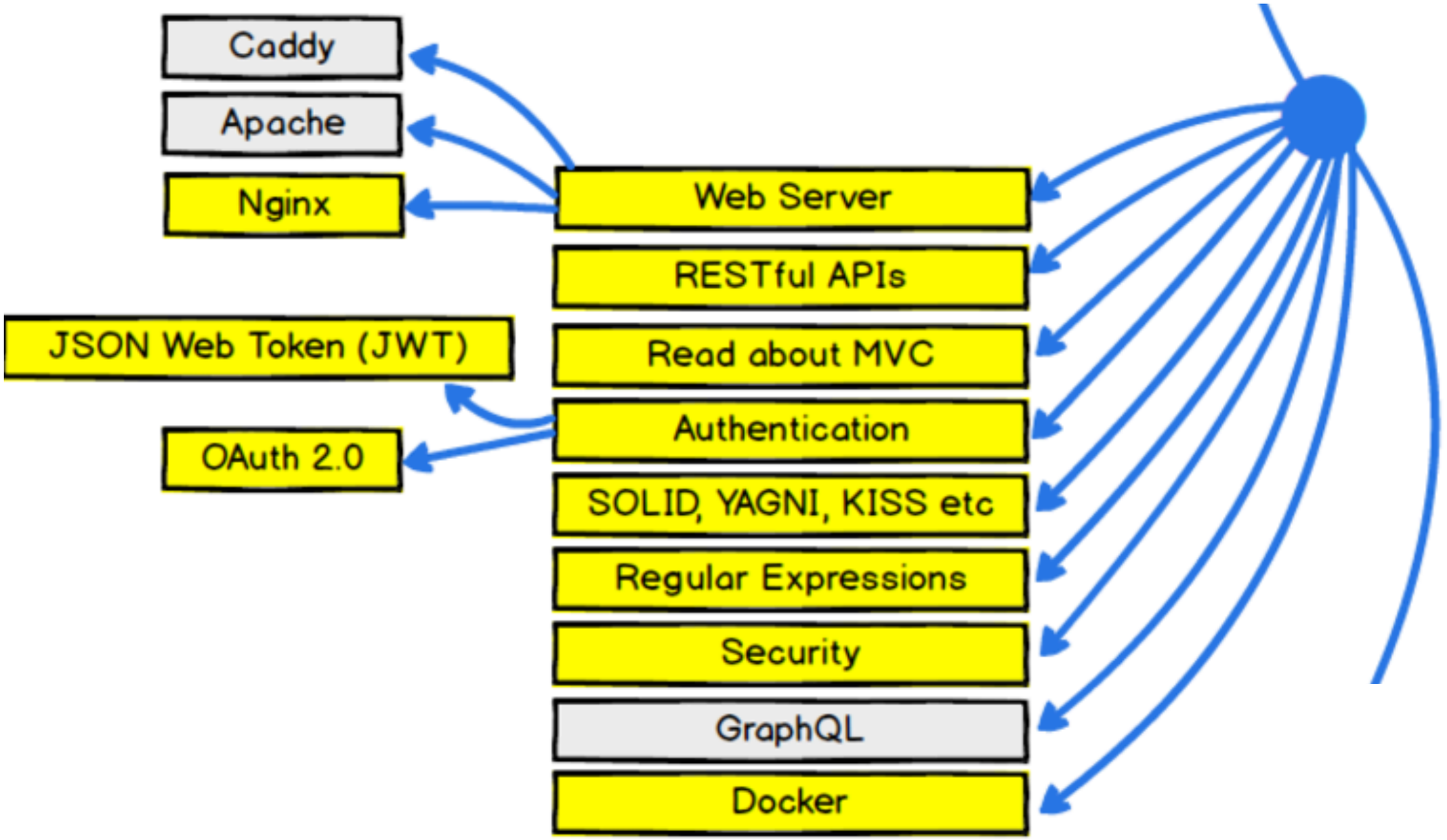
Database

Patterns

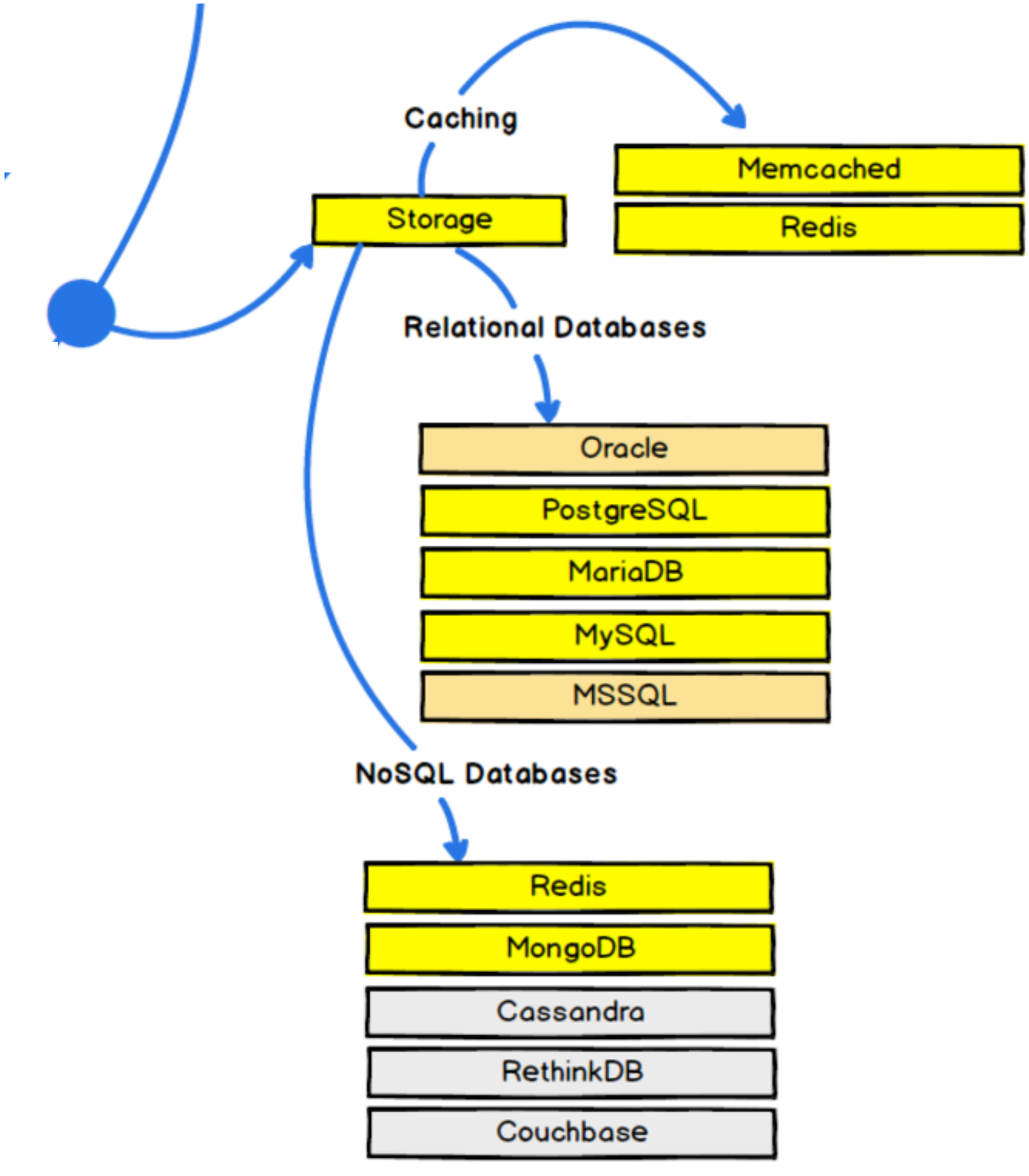
Back-end: Node



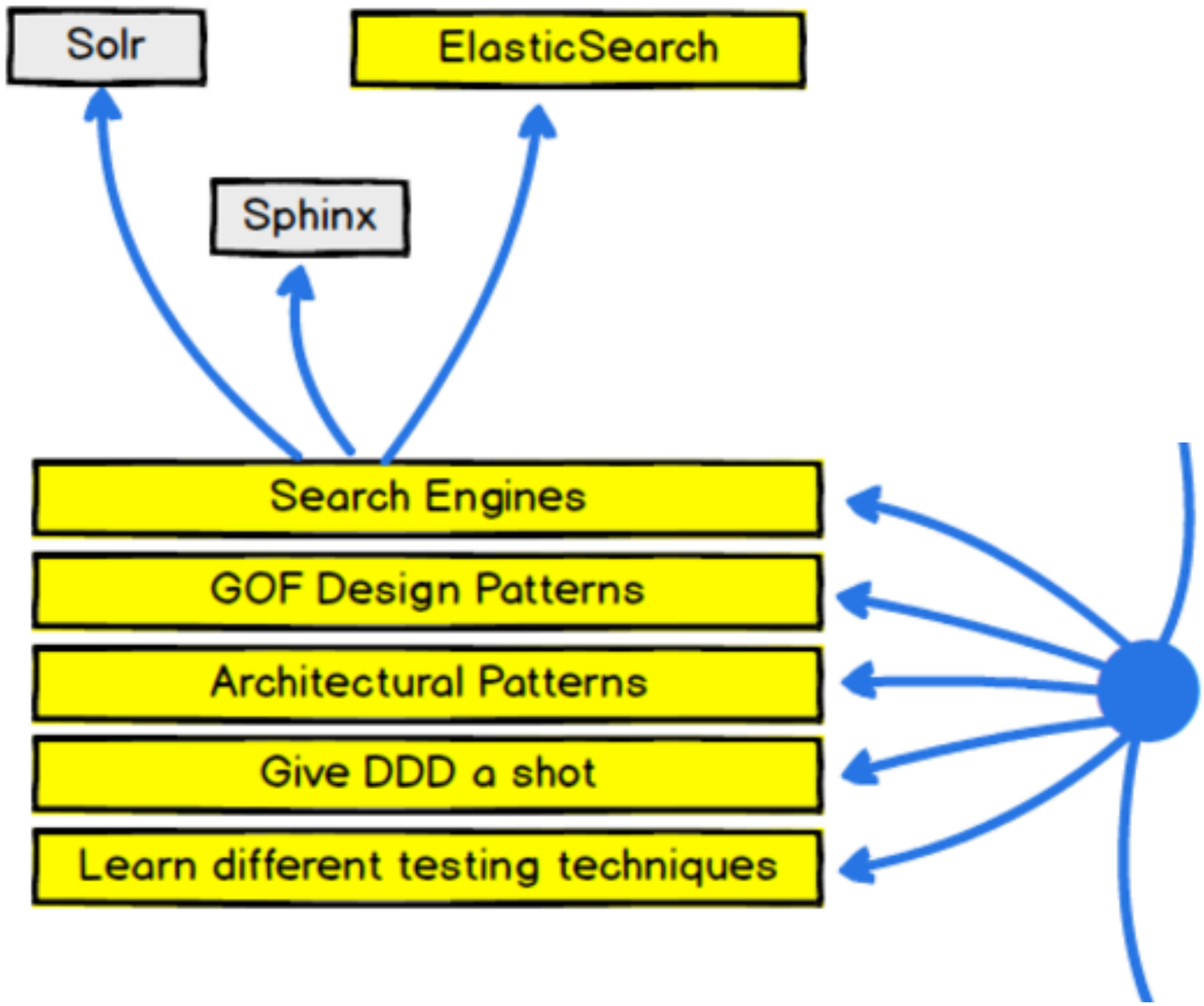
Back-end: Infrastructure + Key Techniques



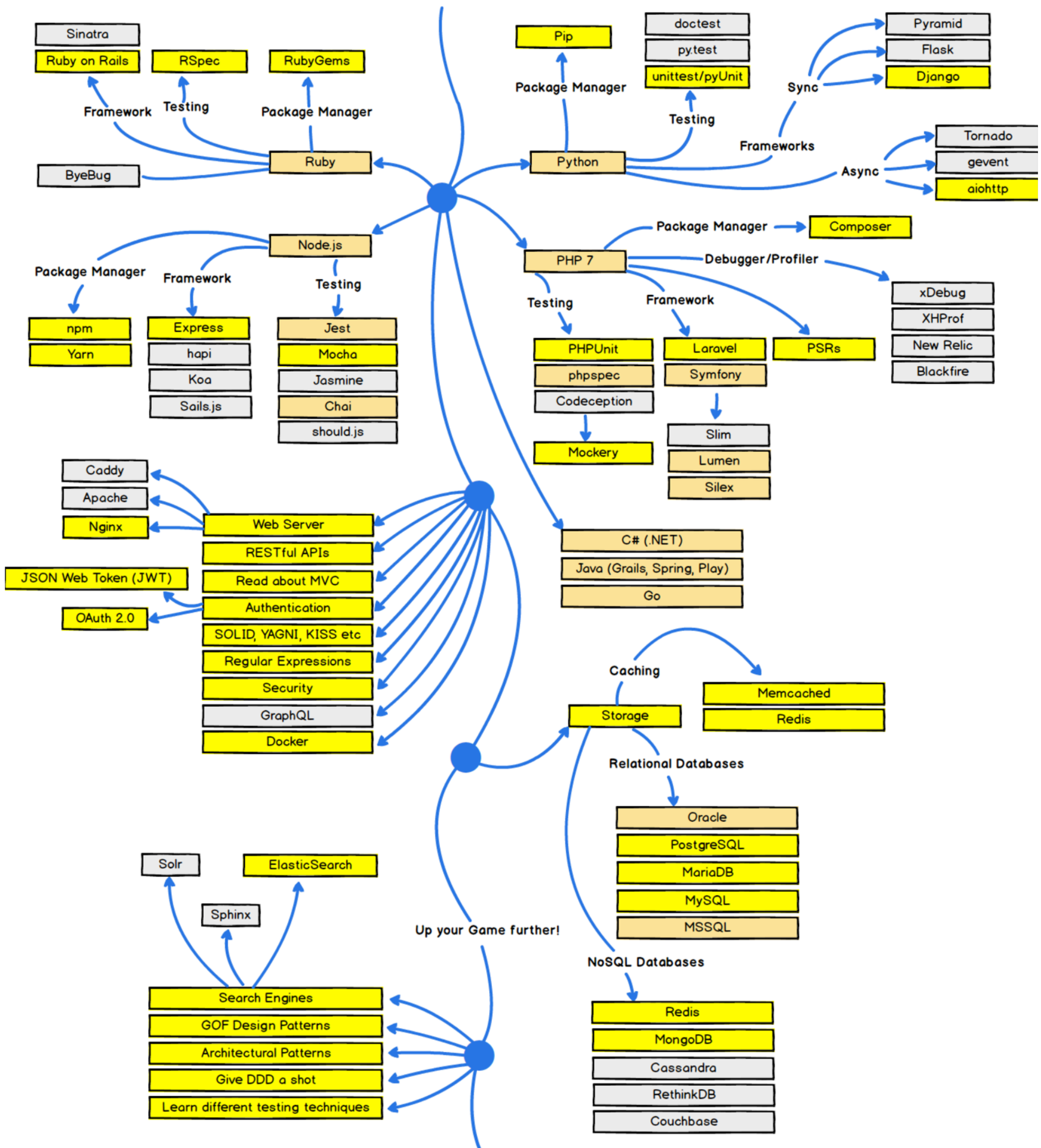
Back-end: Database



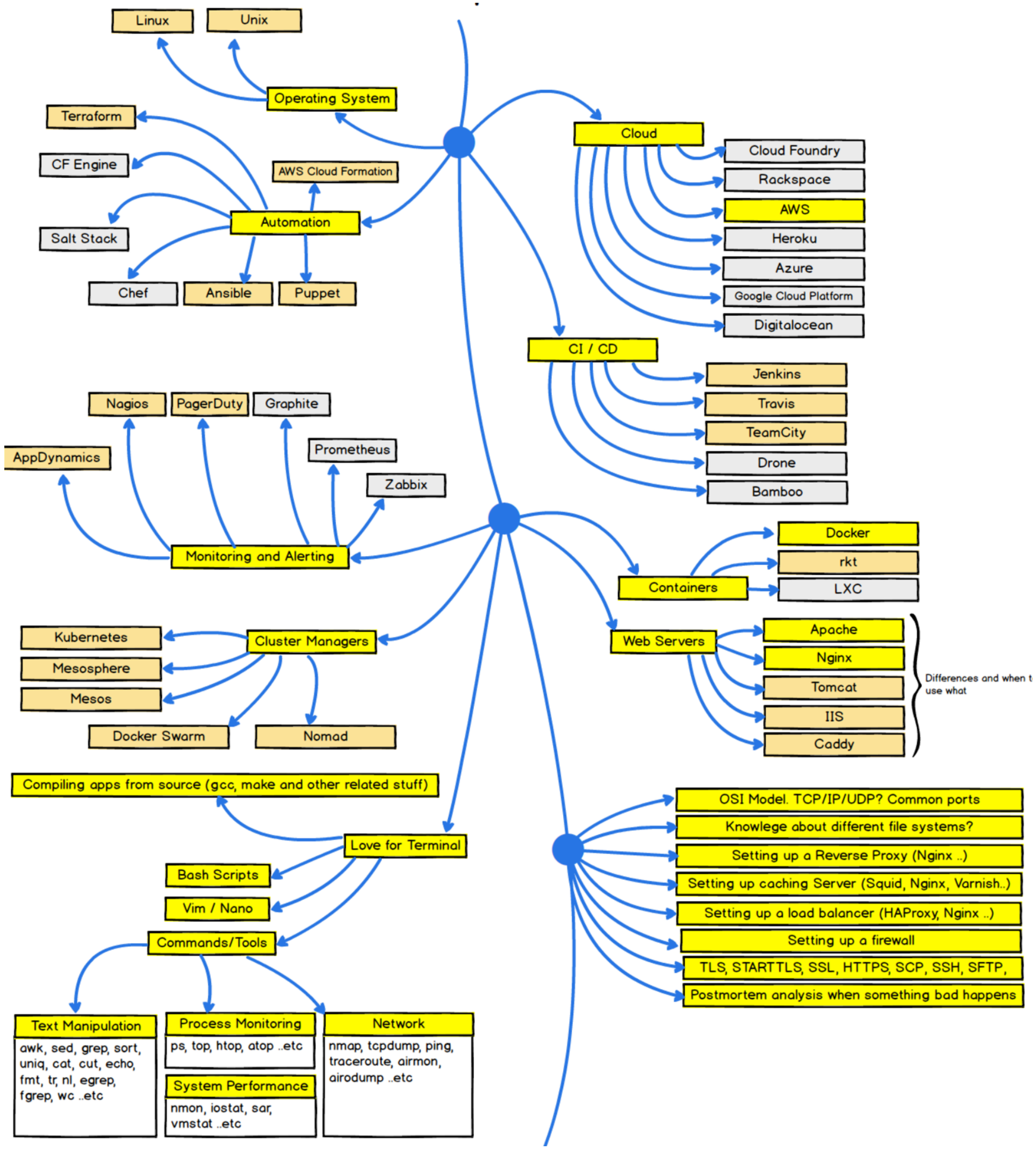
Back End: Patterns



Back End:


Sinatra
Ruby on Rails
RSpec
RubyGems
Framework
Testing
Package Manager
Ruby
ByeBug
Pip
Package Manager
doctest
py.test
unittest/pyUnit
Testing
Sync
Pyramid
Flask
Django
Frameworks
Python
Async
Tornado
gevent
aiohttp
Node.js
Package Manager
Framework
Testing
npm
Yarn
Express
hapi
Koa
Sails.js
Jest
Mocha
Jasmine
Chai
should.js
Package Manager
Composer
PHP 7
Debugger/Profiler
Testing
Framework
xDebug
XHProf
New Relic
Blackfire
PHPUnit
phpspec
Codeception
Mockery
Laravel
Symfony
PSRs
Slim
Lumen
Silex
Caddy
Apache
Nginx
Web Server
RESTful APIs
JSON Web Token (JWT)
Read about MVC
OAuth 2.0
Authentication
SOLID, YAGNI, KISS etc
Regular Expressions
Security
GraphQL
Docker
C# (.NET)
Java (Grails, Spring, Play)
Go
Caching
Storage
Memcached
Redis
Relational Databases
Oracle
PostgreSQL
MariaDB
MySQL
MSSQL
Solr
ElasticSearch
Sphinx
Search Engines
GOF Design Patterns
Architectural Patterns
Give DDD a shot
Learn different testing techniques
Up your Game further!
NoSQL Databases
Redis
MongoDB
Cassandra
RethinkDB
Couchbase

DevOps


Linux
Unix
Operating System
Terraform
CF Engine
AWS Cloud Formation
Automation
Salt Stack
Chef
Ansible
Puppet
Cloud
Cloud Foundry
Rackspace
AWS
Heroku
Azure
Google Cloud Platform
Digitalocean
CI / CD
Jenkins
Travis
TeamCity
Drone
Bamboo
Nagios
PagerDuty
Graphite
AppDynamics
Prometheus
Zabbix
Monitoring and Alerting
Containers
Docker
rkt
LXC
Kubernetes
Mesosphere
Mesos
Cluster Managers
Docker Swarm
Nomad
Web Servers
Apache
Nginx
Tomcat
IIS
Caddy
Differences and when t
use what
Compiling apps from source (gcc, make and other related stuff)
Love for Terminal
Bash Scripts
Vim / Nano
Commands/Tools
Text Manipulation
awk, sed, grep, sort, uniq, cat, cut, echo, fmt, tr, nl, egrep, fgrep, wc ..etc
Process Monitoring
ps, top, htop, atop ..etc
System Performance
nmon, iostat, sar, vmstat ..etc
Network
nmap, tcpdump, ping, traceroute, airmon, airodump ..etc
OSI Model. TCP/IP/UDP? Common ports
Knowlege about different file systems?
Setting up a Reverse Proxy (Nginx ..)
Setting up caching Server (Squid, Nginx, Varnish..)
Setting up a load balancer (HAProxy, Nginx ..)
Setting up a firewall
TLS, STARTTLS, SSL, HTTPS, SCP, SSH, SFTP,
Postmortem analysis when something bad happens