

# Lightning talks #2

2022-10-25

# Lightning Talks ⚡

GBG Development Competence

## Lightning Talk ⚡

*What are Lightning Talks?*

Fast and furious!

- A 30-90 minute session consisting of multiple topics and speakers
- Short talks, aiming for max 5 min
- Concise and to the point

*How to perform Lightning Talks?*

- At least superficial understanding of topic/domain
- Combine main idea and it's most common objection
- Rehearse at least once (5 min) to ensure brevity

Handoff to next talk

- Moonwalk off the stage (optional)



devies  
[www.devies.se/](http://www.devies.se/)

# Today's Agenda

Playcode.io - Sarosh Nasir

Animations in SVGs - Joakim Johansson

Python Development setup - Fredrik Mile

CSR vs SSG vs SSR - Simon Nielsen

Kubernetes node switch/upgrade - Anton Grönlund

# playcode.io



## Javascript Playground

**Code** - start coding within seconds

**Learn** - the easiest way to learn JavaScript/HTML/CSS

**Easy & Fast** - the fastest JavaScript coding experience ever made

GET STARTED

```
script.js x
1 const message = 'Welcome to the new way to code and learn. Fast, Zero Setup, Easy!'
2
3 document.querySelector('#the
4
5 console.log({ message
6
```

# Animations in SVGs

With CSS using keyframes:

<https://playcode.io/988681>

<https://codepen.io/yarre/pen/GRdzWOK>

With SVG AnimateMotion Element:

<https://playcode.io/988669>

<https://codepen.io/yarre/pen/GRdzWep>

# Python Development Setup

Fredrik Mile

# Why?

EXPLORER

...

✓ **BACKEND**

- > .config
- > .husky
- > .vscode
- > ApiTests
- > EmployeeTool.FunctionApp
- > EmployeeTool.Server
- > EmployeeTool.Settings
- > EmployeeTool.Shared
- > Pipelines
- ⚙ .editorconfig
- 🔒 .gitignore
- 🔥 Directory.Build.props
- ≡ EmployeeTool.sln
- 📄 Makefile
- 📖 README.md
- ≡ requirements.txt
- 🔗 very\_important.py



```
tools/Backend <demo*» » python3 very_important.py
```

```
Traceback (most recent call last):
```

```
  File "/Users/fredrikmile/Documents/dev/tools/Backend/very_important.py", line 1, in <module>  
    import requests
```

```
ModuleNotFoundError: No module named 'requests'
```



# Python Virtual Environment

*A virtual environment is a tool that helps to keep dependencies required by different projects separate by creating isolated python virtual environments for them.*

```
backend $ python3 -m venv venv
```

```
backend $ source venv/bin/activate
```

```
backend (venv) $ pip3 install -r requirements.txt
```

```
backend (venv) $ _
```

# Python vs Python3

**python3 very\_important.py**

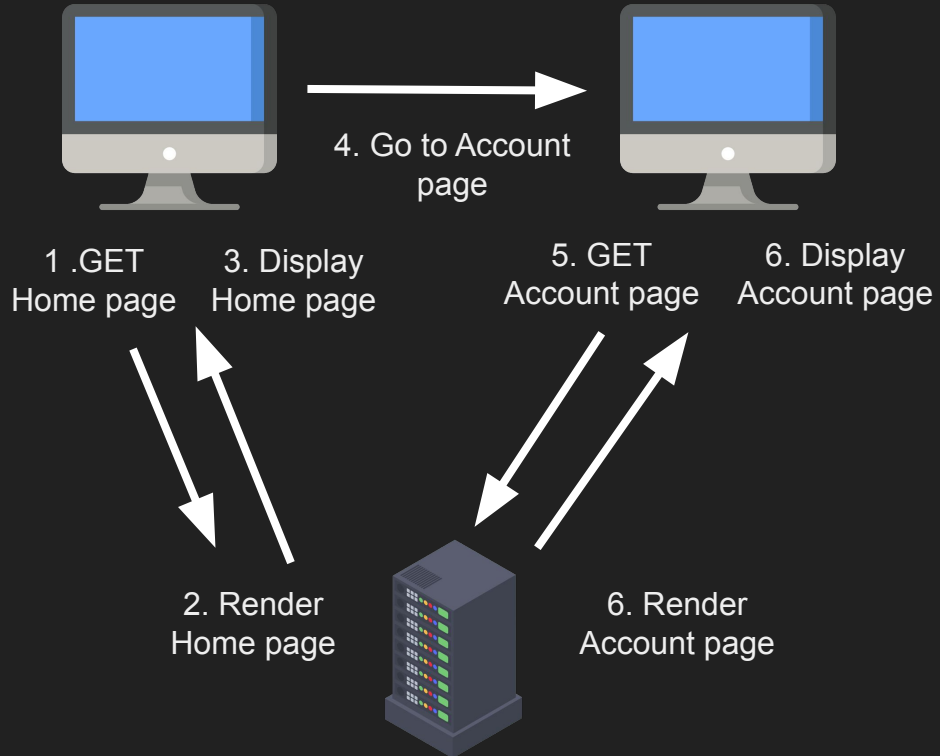
**Instead of**

**python very\_important.py**

# Client-Side Rendering (CSR) vs Server-Side Rendering (SSR) vs Static Site Generation (SSG)

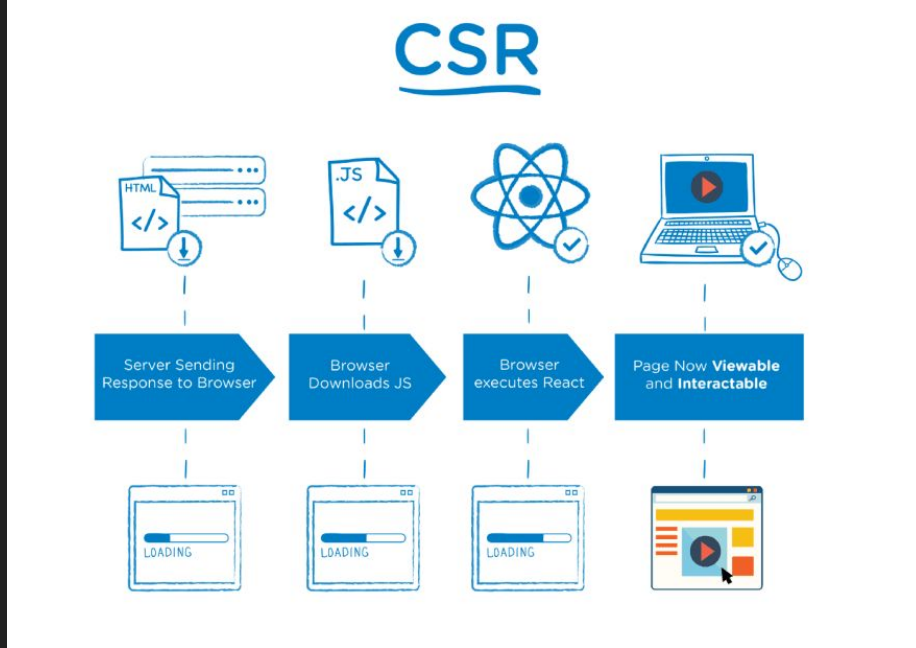
What is it and when to use what?

# Classic Server-side rendering



- Load on each page change
- Can be hard and expensive to scale

# Client-side rendering (Single Page Application)



## Pros:

- Easy to set up (CRA, Vite etc)
- Cheap and easy to host
- Common - fairly easy for new developers

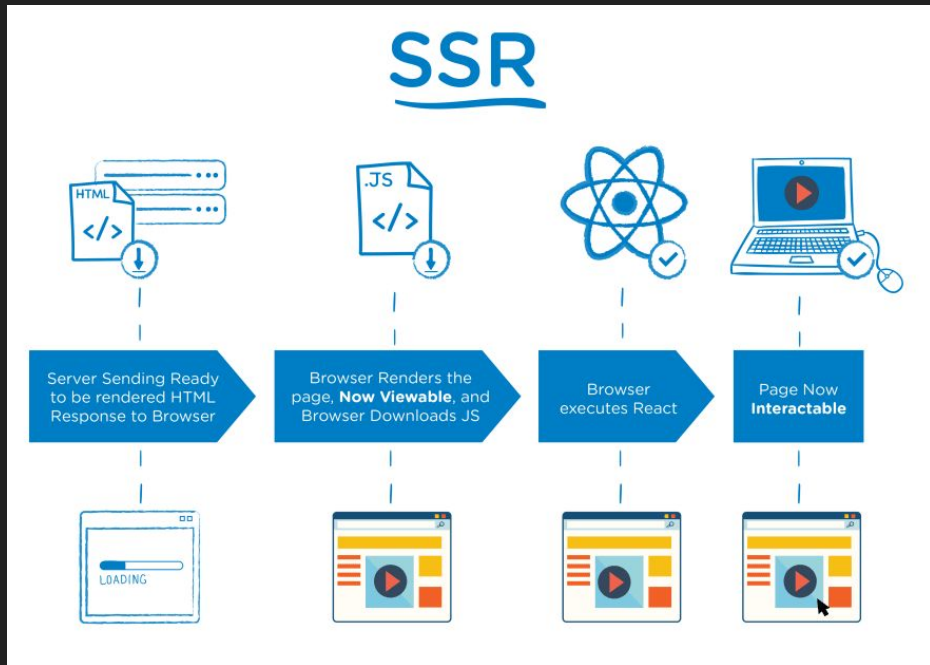
## Cons:

- Initial load can be slow
- Worse for SEO

## Good for:

- Admin UIs. Highly interactive web apps

# Server-side rendering (with Hydration)



## Pros:

- Fairly easy to set up with Next.js
- Fast initial load
- Better for SEO

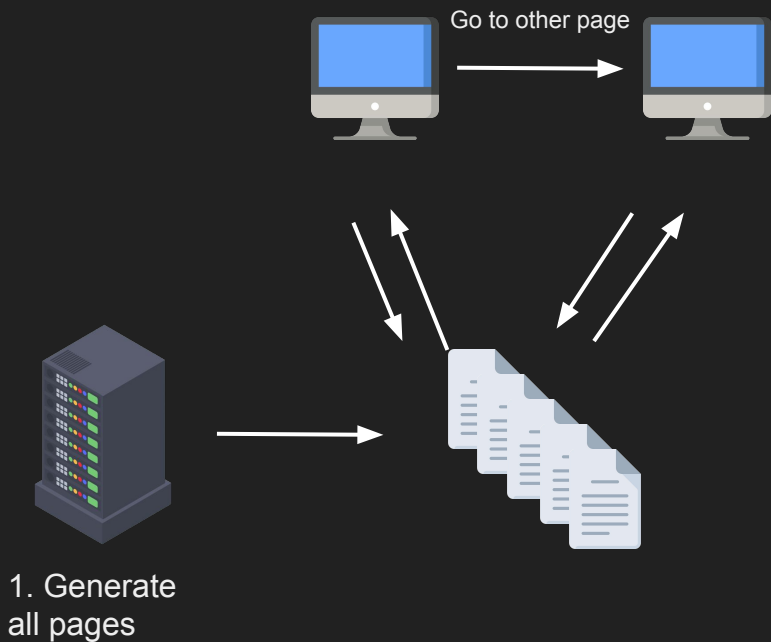
## Cons:

- More expensive and harder to host (especially for scale)
- Steeper learning curve

## Good for:

- Ecommerce and other sites that need good SEO

# Static Site Generation



## Pros:

- Fairly easy to build
- Fast initial load
- Good for SEO
- Cheap and easy to host

## Cons:

- Regenerate all or many pages on change
- Harder to make dynamic content

## Good for:

- Blogs, landing pages and other pages with static content