

# 웹 프론트엔드 바로보기

2016010873 박정욱

# 목차

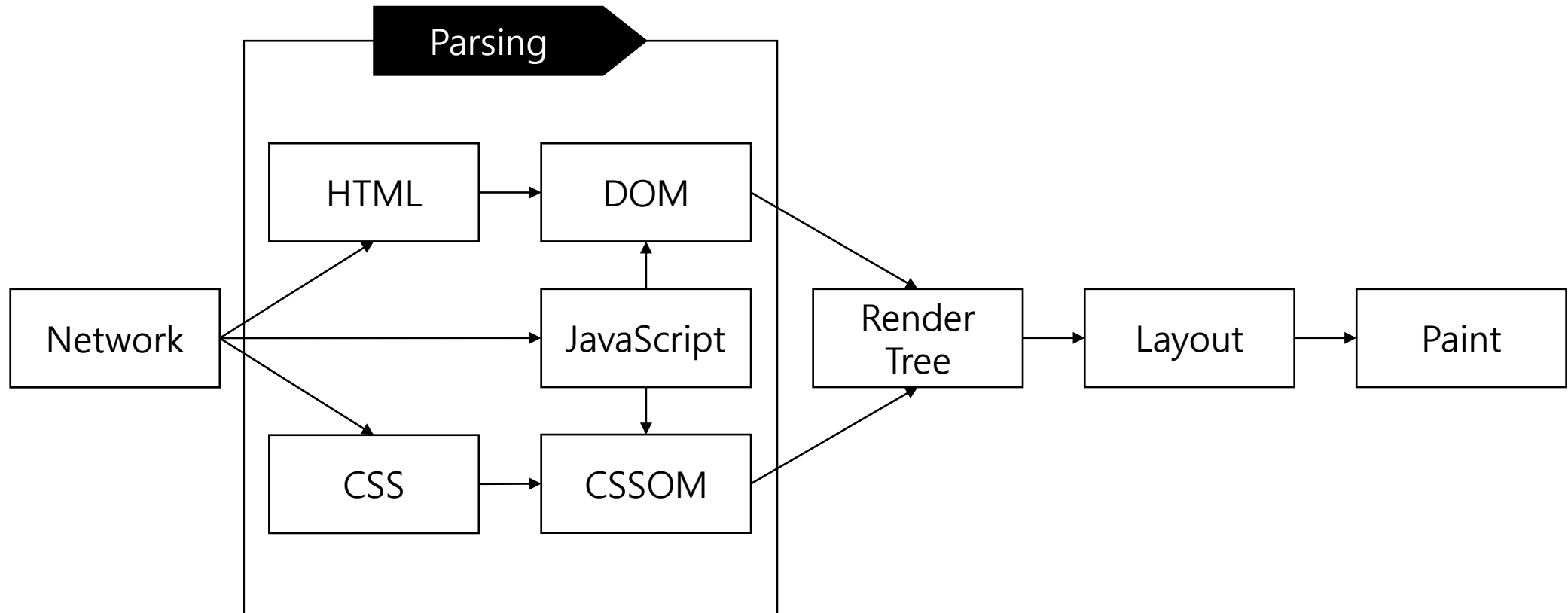
- 브라우저 렌더링 동작 과정
  - 중요 렌더링 경로(Critical Rendering Path, CRP)
  - 문서 객체 모델(Document Object Model, DOM)
  - CSS 객체 모델(CSS Object Model, CSSOM)
- 웹 프론트엔드 개발의 흐름
  - JavaScript의 발전, Client-side -> Server-side
  - MPA -> SPA
  - 명령형 -> 선언형

# 브라우저 렌더링 동작 과정

6.0 ms	0.0 ms	0.0 ms	■ Send Request
6.0 ms	0.0 ms	0.0 ms	■ Receive Response
16.4 ms	0.0 ms	0.0 ms	■ Receive Data
16.5 ms	0.0 ms	0.0 ms	■ Send Request
16.7 ms	0.0 ms	0.0 ms	■ Send Request
17.2 ms	0.0 ms	0.0 ms	■ Finish Loading
17.3 ms	0.3 ms	0.3 ms	■ Parse HTML
17.8 ms	0.0 ms	0.0 ms	■ Receive Response
17.8 ms	0.0 ms	0.0 ms	■ Receive Data
18.1 ms	0.2 ms	0.2 ms	■ Parse Stylesheet
18.2 ms	0.0 ms	0.0 ms	■ Finish Loading
18.4 ms	0.0 ms	0.0 ms	■ Receive Response
18.5 ms	0.0 ms	0.0 ms	■ Receive Data
18.6 ms	0.0 ms	0.0 ms	■ Finish Loading
19.1 ms	0.0 ms	0.0 ms	■ Parse HTML

19.1 ms	0.0 ms	0.0 ms	■ Parse HTML
19.1 ms	0.2 ms	0.2 ms	■ Recalculate Style
19.4 ms	0.1 ms	0.1 ms	■ Layout
19.5 ms	0.0 ms	0.0 ms	■ Pre-Paint
19.5 ms	0.0 ms	0.0 ms	■ Paint
19.6 ms	0.0 ms	0.0 ms	■ Paint
19.6 ms	0.1 ms	0.1 ms	■ Composite Layers
19.7 ms	0.0 ms	0.0 ms	■ Parse HTML
19.8 ms	0.0 ms	0.0 ms	■ Parse HTML
35.9 ms	0.0 ms	0.0 ms	■ Pre-Paint
36.0 ms	0.0 ms	0.0 ms	■ Composite Layers

# 중요 렌더링 경로(CRP)



# 문서 객체 모델(DOM)

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Document</title>
5  </head>
6  <body>
7    <div>
8      <h1>Sample web page</h1>
9      <ul>
10         <li id="item1">Item 1</li>
11         <li id="item1">Item 2</li>
12         <!-- This line is a comment -->
13         <li id="item1">Item 3</li>
14      </ul>
15    </div>
16  </body>
17 </html>
```

HTML Document



```
DOCTYPE: html
HTML
├── HEAD
│   ├── #text:
│   └── TITLE
│       └── #text: Document
│   └── #text:
├── BODY
│   ├── #text:
│   ├── DIV
│   │   ├── #text:
│   │   ├── H1
│   │   │   └── #text: Sample web page
│   │   ├── #text:
│   │   └── UL
│   │       ├── #text:
│   │       ├── LI id="item1"
│   │       │   └── #text: Item 1
│   │       ├── #text:
│   │       ├── LI id="item1"
│   │       │   └── #text: Item 2
│   │       ├── #text:
│   │       ├── #comment: This line is a comment
│   │       ├── #text:
│   │       ├── LI id="item1"
│   │       │   └── #text: Item 3
│   │       └── #text:
│   └── #text:
└── #text:
```

DOM Tree

# 문서 객체 모델(DOM)

```
<div id="container">
  <h1 id="header">Sample web page</h1>
  <ul>
    <li id="item1">Item 1</li>
    <li id="item2">Item 2</li>
    <!-- This line is a comment -->
    <li id="item3">Item 3</li>
  </ul>
</div>
```



```
window.onload = e => {
  document.getElementById("container").style = "background-color: yellow;"
  document.getElementById("header").textContent = "Title has been changed!";
  document.getElementById("item1").remove();
};
```

## Sample web page

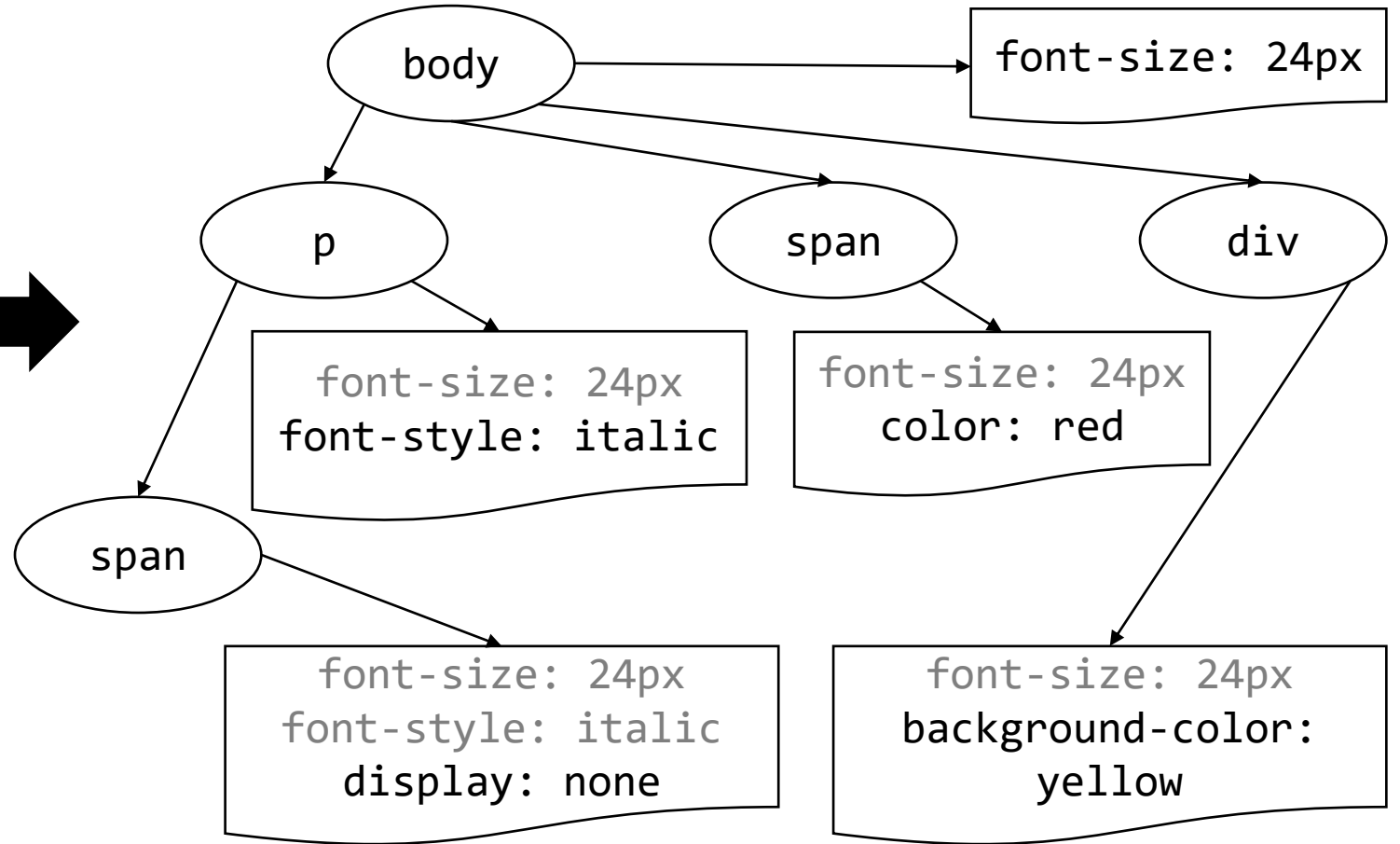
- Item 1
- Item 2
- Item 3

## Title has been changed!

- Item 2
- Item 3

# CSS 객체 모델(CSSOM)

```
body {  
  font-size: 24px;  
}  
  
div {  
  background-color: yellow;  
}  
  
p {  
  font-style: italic;  
}  
  
span {  
  color: red;  
}  
  
p span {  
  font-weight: bold;  
}
```



# 파싱 완료 후

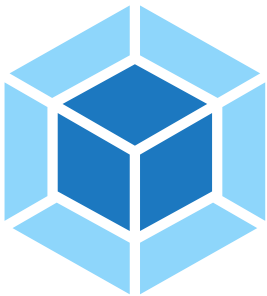
- Render Tree 생성
  - `<script>`, `<meta>` 등 보이지 않는 객체는 제거
  - CSS 규칙(`display: none` 등)을 이용해 숨겨진 객체 또한 제거
  - 이외의 객체에는 각각의 CSSOM 규칙을 적용하여 트리에 추가
- Layout
  - 기기의 viewport에 따른 객체의 정확한 위치와 크기 계산
- Paint
  - 계산이 완료된 객체를 화면에 그리기



# 웹 프론트엔드 개발의 흐름



- JavaScript의 사용 범위 확대
  - Node.js의 등장, 브라우저 밖에서도 JS를
  - Express.js, Electron, etc....



- Module 시스템 정립
  - CommonJS, AMD 등의 모듈 표준 제정
  - webpack, esbuild 등의 번들러 등장

# 웹 프론트엔드 개발의 흐름



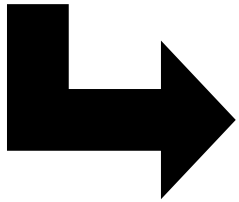
- Multi Page에서 Single Page로
  - 템플릿 기반 SSR -> AJAX 기반 CSR
  - 프론트엔드 분야 전문화



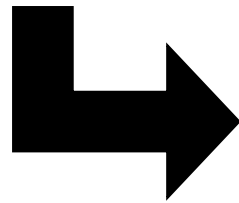
- 프로그래밍 패러다임의 전환
  - 명령형 -> 선언형
  - 절차지향/객체지향 -> 함수형
  - 반응형 프로그래밍

# 웹 프론트엔드 개발의 흐름

Multi Page Application	명령형
HTML, CSS, Vanilla JS	jQuery



JS 용도 범위 확대	Module 시스템 발전
Node.js, webpack, etc....	



Single Page Application	선언형
AngularJS, React, Vue.js	

# 정리

- 브라우저 렌더링 동작 과정
  - 웹 프론트엔드 개발: **DOM**과 **CSSOM**을 조작하여 기능을 개발
  - 중요 렌더링 경로 최적화 또한 프론트엔드 개발자의 몫
- 웹 프론트엔드 개발 흐름
  - JS의 사용 범위 확대에 따른 모듈 시스템 정립 요구
  - AJAX의 인기에 힘입어, MPA보다 UX를 중시하는 SPA 요구
  - 명령형에서 선언형으로 바뀐 UI 라이브러리 트렌드