



# 웹 프로그래밍

## 9. JavaScript 기초 실습 – Play Hangman!

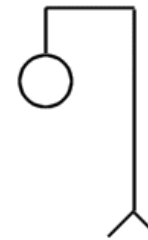
# Hangman Game



Press New Game to play!

Guess

New Game



--- s --- t ---

Guess

New Game

Guesses: ast

# Hangman Game



- playjs.html
  - 전체적인 layout 구성
- playjs.css
  - 페이지 디자인
- playjs.js
  - Hangman 게임 동작을 위한 JS 코드 포함
    - 새로운 게임 시작
      - 단어장에서 워드 랜덤 선택
      - 단어 길이만큼 '\_' 출력
    - 사용자 guess 입력
    - 문자열에서 해당 문자 및 '\_' 출력
- hangman1 ~ 6.gif
  - Hangman 이미지

# playjs.html

- <div id="content"> 다음에 추가

```
<div id="content">
    .....
</div>
<div id="hangman">
    <div>
        
    </div>
    <div id="clue">Press New Game to play!</div>
    <div>
        <input id="hguess" type="text" size="1" maxlength="1" />
        <button id="guessbutton" onclick="guessLetter();"
            disabled="disabled">Guess</button>
    </div>
    <div id="newgamearea">
        <button onclick="newGame();">New Game</button>
    </div>
    <div id="guesssstr"></div>
</div>
```

# playjs.css



```
#hangman {  
  text-align: center;  
}  
  
#newgamearea {  
  margin-top: 2em;  
}  
  
#clue, #guessstr {  
  font-family: monospace;  
  font-size: 2em;  
  padding: 1em;  
}
```



## ● 전역 변수 선언

```
// constants
var POSSIBLE_WORDS = ["obdurate", "verisimilitude", "defenestrate",
    "obsequious", "dissonant", "toady", "idempotent"];
var MAX_GUESSES = 6;

// global variables
var guesses = ""; // all letters guessed so far
var guessCount = MAX_GUESSES; // number of guesses player has left
var word; // random word user is trying to guess
```

- guesses
  - 사용자가 추측한 문자들의 문자열
  - 새로운 문자가 입력될 때 마다, 현재 guesses에 추가됨
- guessCount
  - 사용자에게 주어진 최대 추측 회수 (6번)
  - Word에 포함된 문자를 맞추지 못할 때 마다 1씩 감소
- word
  - 현재 게임에서 선택된 단어



## ● 함수 구현

### ● newGame()

- POSSIBLE\_WORDS에서 랜덤하게 단어 하나 선택 하여 word에 할당
- updatePage() 호출
- <button id="guessbutton" > 활성화

### ● guessLetter()

- 사용자가 입력한 문자에 대해서
- guessCount 계산
- guess 문자열 갱신
- updatePage() 호출

### ● updatePage()

- hangman 이미지 갱신
- Clue 문자열 갱신 후 출력
- guess 문자열 출력
- 게임 승판 결정

# newGame()



- newGame()
  - POSSIBLE\_WORDS에서 랜덤하게 단어 선택 후 word에 할당
    - Math 객체의 random() 메소드 활용
    - Math.random() \* 10 => 0에서 9사이의 랜덤값 생성
    - (Math.random() \* 10) + 1 => 1에서 10사이의 랜덤값 생성
  - guessCount와 guesses 초기화
  - <button id="guessbutton" > 활성화
    - disabled 속성에 false 할당
  - updatePage() 호출



# updatePage() – v1

## ● updatePage()

### ● Update clue string

- html 문서의 <div id=clue> 영역
- word의 길이만큼 ‘\_’ 문자 출력
- 추 후, 사용자 입력에 따라서 clue 문자열 갱신

### ● Update guess string

- html 문서의 <div id=guesses> 영역
- 사용자가 지금까지 입력한 추측 문자열 출력

### ● Update hangman image

- Html문서의 <img> 영역
- guessCount에 따른 이미지 갱신
- 예) guessCount가 3이면 <img src = hangman3.gif>

# guessLetter()

## ● 사용자 신규 입력 문자

- <input id=hguess >를 통해 문자 입력
- 변수 letter에 사용자 입력 문자 할당
- letter가 이미 guesses에 포함된 문자라면 return
- 새로운 문자라면
  - guesses에 letter 문자 추가
  - guessCount 갱신
    - word에 letter가 포함되어 있지 않을 경우에만 감소
    - 문자열 메소드인 indexOf() 활용
      - str.indexOf('e') => str 문자열에서 문자 'e'가 위치한 인덱스 반환
      - 문자열이 없으면 음수 값 반환
  - updatePage() 호출

# updatePage() - final



- Clue 문자열 갱신
  - 사용자 입력이 추가된 guesses에 대해서
  - Word에 포함된 문자는 해당 위치에 문자 출력하고 나머지는 '\_' 출력
  - 문자열 메소드 indexOf() 와 charAt() 활용
    - str.charAt(i) => str 문자열에서 i번째 인덱스에 위치한 문자 반환
- Guess 문자열 출력 영역 (id=guesstr 영역) 갱신
  - GuessCount == 0, "you lose" 출력
  - Clue string에 '\_'가 하나도 없으면, "you win" 출력
  - 그 이외에는 guesses 문자열 출력