

2020. 07. 04
학습내용 발표

알고리즘 - N진수 게임

문제

튜브가 활동하는 코딩 동아리에서는 전통적으로 해오는 게임이 있다. 이 게임은 여러 사람이 둥글게 앉아서 숫자를 하나씩 차례대로 말하는 게임인데, 규칙은 다음과 같다. 숫자를 0부터 시작해서 차례대로 말한다. 첫 번째 사람은 0, 두 번째 사람은 1, ... 열 번째 사람은 9를 말한다. 10 이상의 숫자부터는 한 자리씩 끊어서 말한다. 즉 열한 번째 사람은 10의 첫 자리인 1, 열두 번째 사람은 둘째 자리인 0을 말한다.

이렇게 게임을 진행할 경우,

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 0, 1, 1, 1, 2, 1, 3, 1, 4, ...

순으로 숫자를 말하면 된다.

한편 코딩 동아리 일원들은 컴퓨터를 다루는 사람답게 이진수로 이 게임을 진행하기도 하는데, 이 경우에는

0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, ...

순으로 숫자를 말하면 된다.

이진수로 진행하는 게임에 익숙해져 질려가던 사람들은 좀 더 난이도를 높이기 위해 이진법에서 십육진법까지 모든 진법으로 게임을 진행해보기로 했다. 숫자 게임이 익숙하지 않은 튜브는 게임에 저서 벌칙을 받는 굴욕을 피하기 위해, 자신이 말해야 하는 숫자를 스마트폰에 미리 출력해주는 프로그램을 만들려고 한다. 튜브의 프로그램을 구현하라.

입력 형식

진법 n , 미리 구할 숫자의 갯수 t , 게임에 참가하는 인원 m , 튜브의 순서 p 가 주어진다.

$$2 \leq n \leq 16$$

$$0 < t \leq 1000$$

$$2 \leq m \leq 100$$

$$1 \leq p \leq m$$

출력 형식

튜브가 말해야 하는 숫자 t 개를 공백 없이 차례대로 나타낸 문자열. 단, 10~15는 각각 대문자 A~F로 출력한다.

```
var every = [];
for(var i=0; i<t*m; i++){
    every.push(i.toString(n)) // 진법변환
}
//다 이어서 문자열로 만들기
var add = every.join("");

// 말해야 하는 숫자 출력하기
for(var j=0; j<t*m; j++){
    if(j%m===p-1){ // 몇번째인지 조건 구하기 (index가 0부터 시작하므로)
        answer = answer + add[j].toUpperCase(); //A~F 때문에
    }
}
```

JavaScript for Web Browser (생활코딩)

2020.07.04

jQuery Object – map()

map 메소드는 제공된 배열 데이터를 변환시켜 다시 새로운 배열 객체를 반환(returns a new array)한다.

.map(array, callback(elementOfArray, indexInArray))

- array : 변환할 배열
- callback(elementOfArray, indexInArray) 모든 요소에 대해 실행될 함수.

첫번째 인자는 배열 요소(item), 두번째 인자는 배열 인덱스

.map(callback(index,elem))

콜백함수의 this는 현재 DOM를 참조합니다. 함수는 각각의 데이터 값을 가지고 오며, 데이터 값은 배열에 저장됩니다. 만약 null이나 undefined 값이 나온다면 각각의 데이터 값을 제대로 가지고 오지 않은 것입니다.

```
var li = $('li');  
li.map(function(index,elem){  
    $(elem).css('color','red');  
})
```

리액트를 다루는 기술

props 검증 : propTypes

컴포넌트의 필수 props를 지정하거나 props 타입(type)를 지정할 때는 propTypes를 사용합니다.

```
import PropTypes from 'prop-types';
```

코드 위쪽에 propTypes를 불러옵니다.

```
MyComponent.propTypes={  
  name : PropTypes.string,  
  age : PropTypes.number.isRequired  
}
```

isRequired를 이용하여 필수 propTypes를 지정합니다. 해당 부분의 prop값이 입력되지 않으면 오류를 출력합니다.

propTypes 종류

array / bool / func / number / object / string

symbol – ES6 문법의 심벌 개체

node - 렌더링 할 수 있는 모든 것 (숫자, 문자열, element 또는 이들로 구성된 배열)

element – 리액트요소

instanceOf(Myclass) – 특정 클래스의 인스턴스

oneOf(['Male', 'Female']) – 주어진 배열 요소 중 하나

oneOfType([React.PropTypes.string, React.PropTypes.number]) : 주어진 배열 안의 종류 중 하나

arrayOf(React.PropTypes.number) : 주어진 종류로 구성된 배열

objectOf(React.PropTypes.number) : 주어진 종류의 값을 가진 객체

shape({name : React.PropTypes.string, age : React.PropTypes.number}) : 주어진 스키마를 가진 객체

any : 아무종류

리액트를 다루는 기술

리액트의 이벤트 시스템

이벤트 사용시 주의사항

1. 이벤트 이름은 camelCase로
2. 이벤트에 실행할 자바스크립트 코드를 전달하는 것이 아니라, 함수형태의 값을 전달
3. DOM 요소에만 이벤트를 설정할 수 있음

```
<input
  type="text"
  name="message"
  placeholder="아무거나 입력해보세요"
  value={this.state.message}
  onChange={this.handleChange}
  onKeyPress={this.handleKeyPress}
/>
```

```
handleKeyPress = (e) => {
  console.dir(e.keyCode)
  if(e.keyCode === 13 ){
    this.handleClick();
  }
}
```

```
handleKeyPress = (e) => {
  if(e.key === 'Enter' ){
    this.handleClick();
  }
}
```