**[직무부트캠프] 프론트엔드**

**2주차 과제 – 안유리**

|  |  |
| --- | --- |
| **프론트 기본 개념 내용 정리** | |
| **1. HTML의 form과**  **action의 개념** | **- method / GET & POST / parameter (name, value)**  + 서버는 정보를 제공하면 클라이언트가 정보를 제공 받음  + Form 이란? 사용자의 데이터를 서버에 전송하는 방법  ex) 로그인 시, 아이디/비밀번호 입력할 때, 블로그에 글 작성할 때, 파일을 전송할 때 등  **<form> 태그**  텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다.  1. **name**: form의 이름 결정  2. **action**: 사용자가 입력한 데이터를 받아 처리하기 위한 웹 프로그램 페이지 지정  텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다.3. **method**: 웹 서버와 클라이언트 간의 통신 방법 지정  (1) **GET 방식** (default)  - URL 뒤에 파라미터를 붙여서 데이터를 전달하는 방식  - 사용자가 보내는 데이터는 이름과 값이 & 기호로 결합된 문자열 형태로 전달됨  - 서버로 보낼 수 있는 최대 글자수 2,048자  - URL을 보면 어떤 데이터를 전송하고자 하는지 알 수 있기에 보안에 취약  → 데이터의 크기가 작고, 중요하지 않은 정보를 보낼 때 주로 사용    텍스트, 소프트웨어, 웹 페이지, 웹사이트이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다.  텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다.(2) **POST 방식**: HTTP Request 헤더에 파라미터를 붙여서 데이터를 전송하는 방식  - 서버로 보낼 수 있는 글자수 제한 없음  - GET 방식과 비교하여 보안상 우위에 있음  - 다량의 데이터를 전송할 때 사용  텍스트, 소프트웨어, 스크린샷, 웹 페이지이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다.  4. **type**: form의 모양과 기능 설정  **<Parameter>**  - name(이름)과 value(값)으로 구성  - 클라이언트에서 요청 시 넘어가는 데이터  - ‘?’ 가 파라미터의 시작을 의미, ‘파라미터 이름=파라미터 값’ 으로 작성  - 주소 뒤의 “?이름=값” 을 쿼리스트링 이라 부름  텍스트, 소프트웨어, 웹 페이지, 웹사이트이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다.- ex) 아래의 주소에서는,  where=nexearch  sm=top\_sly.hst  fbm=0  acr=6  ie=utf8  query=프랭크버거  ackey=0bohn4b5 의 총 7개의 파라미터가 & 기호로 연결되어 있음 |
| [참고 자료]  https://cafe.naver.com/codingstudyplace/18  https://www.w3schools.com/tags/att\_form\_method.asp  https://lasbe.tistory.com/83#02 |
| **2. Javascript** | **- DOM Manipulation**  **<DOM (Document Object Model)>**  - HTML, XML 문서를 프로그래밍 언어에서 조작할 수 있도록 객체 형태로 표현한 것  - 웹 페이지에서는 DOM을 이용하여 HTML 요소 조작, event처리 가능  **<Node>**  - DOM 트리는 HTML 태그 요소, 속성, 텍스트, 주석 등이 트리 형태로 구성됨 → 이들이 각각 **노드**  - DOM 트리의 가장 꼭대기에 있는 노드가 **루트 노드** → html 이 루트 노드 (document는 객체)  - 각 노드는 부모, 자식, 형제 관계가 성립됨 ex) html은 head와 body의 부모 노드  **<요소 선택 (get 메서드)>**  텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다. (1) getElementByID  element = document.getElementByID(id);  - id명과 일치하는 element 객체 반환  - 일치하는 element가 없다면 null 반환  (2) getElementByClassName  elements = element.getElementsByClassName(names);  - class명과 일치하는 element들의 HTMLCollection을 반환  - 일치하는 element가 없다면 빈 HTMLCollection을 반환  + HTMLCollection: 유사 배열 객체 (실제 배열은 아니므로 forEach()같은 함수 사용 불가)  텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다.  (3) getElementsByTagName : 태그명과 일치하는 요소 노드를 모두 선택  **<요소 선택 (query 메서드)>**  (1) querySelector  매개변수로 넘어오는 CSS 선택자에 해당하는 노드를 1개만 선택  (2) querySelectorAll  매개변수로 넘어오는 CSS 선택자에 해당하는 노드를 모두 선택  ✅ query 메서드는 매개변수로 CSS 선택자를 전달받기 때문에 get 메서드보다 범용성이 좋음  **<event 처리 방법>**  스크린샷, 소프트웨어, 텍스트, 멀티미디어 소프트웨어이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다.ex) btn이 클릭되었을 때, click 함수가 실행  (1) addEventListener  <노드>.addEventListener(“<이벤트 타입>”, <이벤트 함수>); 로 작성  - 여러 개의 콜백 함수 지정이 가능하므로, 여러번 사용하더라도 모든 핸들러가 정상적으로 작동  - 캡쳐링, 버블링 같은 이벤트 처리 방법 제어 가능  텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다. (2) onclick  - 마우스로 클릭하면 발생  - onclick 이벤트 핸들러를 여러번 사용하게 되면,  가장 나중에 선언한 이벤트 핸들러가 작동됨  **<EventLoop>**  - 자바스크립트는 한 번에 하나의 작업만 수행할 수 있음  하지만, 웹에서는 여러 작업을 동시에 처리해야 할 때가 많다  → 이를 가능하게 해주는 것이 이벤트 루프(EventLoop)! |
| [참고 자료]  https://cafe.naver.com/codingstudyplace/18  https://seokzin.tistory.com/entry/JavaScript-getElementById%EC%99%80-querySelector-%EC%B0%A8%EC%9D%B4%EC%A0%90  https://www.w3schools.com/jsref/met\_document\_getelementsbyclassname.asp |
| **3. Javascript 문법** | **<var / let / const 차이>**  (1) var (변수)  - 값이 변하는 데이터를 저장하고 관리하기 위한 공간 ex) var num = 10 + 20;  - 중복 선언과 재할당 가능 → 마지막에 할당된 값이 최종으로 변수의 값으로 저장됨  (2) let: 중복 선언 불가 → SyntaxError 발생 ex) let num = 10 + 20;  (3) const: 중복 선언 불가, 재할당 불가  **<ES5의 Array 관련 메소드>**  (1) isArray()  - 체크 대상이 배열이면 true, 아니면 false를 반환  (2) map()  - 배열의 각 요소에 대하여, 순차적으로 주어진 함수를 실행한 반환 값을 모아 새로운 배열 반환  - Callback 함수에서 값을 변경하지 않는 이상, 기존 배열 값은 변경되지 않음  (3) filter()  - 주어진 함수가 true를 반환하는 경우의 element만 포함하는 새로운 배열을 생성  - 원본 배열은 변경되지 않음  (4) forEach()  - 배열에 있는 각각의 요소에 대해 한 번씩 callback 함수 실행  - 함수 내에서 변경하지 않는 이상, 기존 배열 값은 변경되지 않음  **<rest parameter>**  - 매개변수 이름 앞에 세개의 점 **...** 을 붙여서 정의한 매개변수  - 함수에 전달된 인수들의 목록을 배열로 전달받음  - 먼저 선언된 파라미터에 할당된 인수를 제외한 나머지 인수들이 모두 배열에 담겨 Rest 파라미터에  할당됨 → 따라서 **Rest 파라미터는 반드시 마지막 파라미터**가 되어야 함  - 함수 정의 시 선언한 매개변수 개수를 나타내는 함수 객체의 length 프로퍼티에 영향 주지 않음  스크린샷, 텍스트, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다.  **<Arrow function>**  - 파라미터가 하나면 소괄호 생략 가능  - return이 할 줄 이라면 중괄호와 return 생략 가능  **<1급 객체>**  일급객체(First-class Object)란? 다른 객체들에 일반적으로 적용 가능한 연산을 모두 지원하는 객체  - 일급객체의 조건  (1) 변수에 할당(assignment) 할 수 있다  (2) 다른 함수를 인자(argument)로 전달 받는다  (3) 다름 함수의 결과로서 return될 수 있다  **<Object의 개념>**  객체란? 이름(name, key)과 값(value)으로 구성된 프로퍼티의 집합  텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다. - 객체의 프로퍼티 참조  텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진  AI가 생성한 콘텐츠는 부정확할 수 있습니다. (1) object.property (Dot Notation)  - 어떠한 객체의 Key 값을  정확히 알고 있을 때 사용 가능  (2) object[‘property’] (Bracket Notation)  - 객체 사용법  (1) 추가 – obj[“pig”] = “oink” 로 추가  (2) 삭제 – delete obj.cat; 으로 삭제  (3) 수정 – obj.cat = “mew” (기존 객체의 key 값에 새로운 value 값 선언)  (4) 조회 – “cat” in obj →boolean 값으로 결과 return |
| [참고 자료]  https://velog.io/@dongjun187/javascript-var-let-const-%EC%9D%98-%EC%B0%A8%EC%9D%B4%EC%A0%90  https://paperblock.tistory.com/59 https://poiemaweb.com/es6-extended-parameter-handling  https://velog.io/@reveloper-1311/%EC%9D%BC%EA%B8%89-%EA%B0%9D%EC%B2%B4First-Class-Object%EB%9E%80 |