이화이언 웹개발 2차 수습과제

1. CSS
2. 드림코딩 – selector

CSS : Cascading(연속해서 떨어지는) Style Sheet

Author style > User style > Brower (우선순위) -> !important (우선순위 무시, 최대한 쓰지 X)

Selectorts : \*(모든 테그), Tag(해당 테그), #id(해당 아이디), .class(class), :(state), [](attribute)

실습) (jsbin.com) selector { property: value }

Ex. Button:hover { color: red; } //버튼에 마우스를 올려둘 때 색상이 red로 바뀜

(dev 보기 : cmd+option+I)

Padding(위,아래,옆 따로따로 가능) : content 안에 들어가는 여백

margin : content 밖에 들어가는 여백

(실습 게임 : <https://fluckeout.github.io/>)

1. 드림코딩 – display, position

div – block level

span (내용이 있어야 표기) – inline level

* Display에서 block/inline-block로 변경가능

Position : relative(원래 있던 자리)/absolute(담겨 있는 상자)/fixed(존재하는 페이지)/skicky(원래 있던 자리에 있지만 스크롤해도 그자리에 있음) 로 변경해야 위치를 옮길 수 있음

1. 드림코딩 – flexbox

Float : 이미지와 text를 어떻게 배치할지

* Container/item 에 각각 지정되는 속성값 존재
* 중심축(main axis)/반대축(cross axis) 존재 : 해당하는 수직/수평축이 달라짐

실습) height: 100% : 부모 높이의 100% / 100vh : 보이는 디폴트의 100%

(color tool)

-container) display: flex;(왼쪽부터 정렬)

Flex-direction : row(기본), row-reverse(오른쪽부터 정렬) -> 중심축: 수평축

Column(위부터 정렬), column(아래부터 정렬) -> 중심축: 수직축

Flex-wrap : nowrap (기본,모든 item들이 한 줄 존재), wrap(한 줄이 다 차면 다음 줄로 넘어감), wrap-reverse(아래서부터 wrapping)

Flex-flow : direction과 wrap를 순차적으로 한번에 작성 가능

Justify-content(item들을 중심축에서 어떻게 배치할지) : flex-start(기본), flex-end(순서는 유지, 반대쪽에 배치), center(중심), space-around(box 양 옆에 box마다 특정 공간 마련), space-evenly(box 양 옆에 일정한 특정 공간 마련), space-between(맨 끝을 제외한 box 양 옆에 특정 공간 마련)

Align-items(item들을 반대축에서 어떻게 배치할지) : baseline(text가 균일하게 배치),

Aling-content: space-between(맨 끝을 제외한 줄 사이에 특정 공간 마련)

Item) order (순서 변경)

Flex-grow: 0(기본, 크기를 유지하려 하지만, 작아진다면 줄 크기에 맞춰 작아짐), 1(특정 비율에 따라 item의 크기를 변경해서 그 줄을 모두 채움)

Flex-shrink: 0(기본), 1(특정 비율에 따라 item크기가 줄어듦)

Flex-basis: -% (특정 비율에 따라 크기 정하기)

Aling-self : center(특정 item만 상대축 중앙에 배치)

(실습 게임: flexbox froggy)

1. 드림코딩 – 그리드

Css layout) flex : 1차원 / grid : 2차원

실습) Display: grid

Grid) Grid-template-columns / Gird-template-rows : repeat() 사용가능-%, -ft으로 크기 조정 가능

Gird-auto-rows : 자동으로 설정, minmax(최소값은 고정, text가 많아지면 커짐)

Grid-gap(item사이에 gap 배정)

Gird-template-areas : grid cell 마다 이름을 지정해줌 + 각각 item/image에 이름 표시

Grid cell)grid-column(row)-start/end : (차지하는 cell 영역 조정)

Gird-column: 시작숫자 / 끝 숫자(span 크기) 로도 표시 가능

1. JavaScript
2. INTRODUCTION

모멘텀!!, HTML과 CSS에 대한 기본 지식은 필요,

자바스크립트는 프론트엔드를 하는 유일한 언어 + 정말 다양한 곳에서 사용( 백엔드, 3d, 어플, 머신러닝, 실시간 채팅 등등)

1. WELCOME TO JAVASCRIPT

Js파일과 css파일을 브라우저에서 사용하기 위해서는 html을 이용

Js에서의 데이터 타입

-number: int(정수), float

-string: “”를 사용, +로 이어서 사용 가능

Console.log = print

변수) const(변경 불가능) ex. const a = 5;

/ let(변경 가능,생성할 때) 이후 변경할 때에는 그냥 (변수이름) = (변경내용);

Boolean : true/false(값 존재),

null(값이 없음, 자연적으로 발생하지 않음),

undefined(메모리 안에는 존재하지만 정의되지 않음, 자연적으로 발생, 비어있음)

데이터를 그룹하고 싶다면? Array!

Const (배열이름) = [ , , , , ,]; , 인덱스는 0부터 시작

배열이름.push : 새로운 항목 추가

Object 생성 : const (이름) = { 항목이름:값, 항목이름:값 };

이후 이름.항목이름 / 이름[“항목이름”] 으로 표시 가능 (새로운 항목도 바로 추가 가능)

함수 생성 : function (이름)(arguments){ 함수 내용 };

함수를 통해 결과값을 받기 원한다면? Return!

Const (변수이름) = (해당함수); -> 이때 함수에는 return (결과값)이 있어야 함

Prompt(메세지) : 창을 띄워 값을 입력받을 수 있음

parseInt(15) : string 15를 int 15로 변환 -> 숫자 크기 비교 가능 ,

만일 숫자로 바꿀 수 없다면 NaN라고 표시

isNaN(arg): arg가 숫자라면 false, 숫자가 아니라면 true

조건문: if(condition){ (true일 때)} else if (condition2) {(true일 때) } else { (false일 때) }

1. JAVASCRIPT ON THE BROWSER

Document : 본래 존재하는 object

JavaScript는 html을 읽고, 변경 가능

querySelector : element를 css방식으로 검색, 첫번째 element만 가져옴, 모두 가져오기를 원할 시 querySelectorAll 사용

on-- : event -> 모든 event를 javascript에서 listen 가능

ex. title.addEventListener(“click”);

1.element를 가져온다. 2.event를 만든다. 3.event를 listen한다.

1. LOGIN

Event -> subnmit

Event.preventDefault(); : 브라우저의 기본 행동을 막기 -> submit 후 새로고침 막기

Username을 입력 받은 후-> 입력창은 숨기고 username은 저장