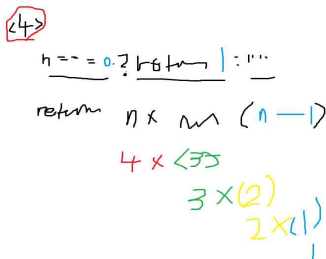


2022 퍼블리셔 부트캠프 020

- JavaScript 함수선언 정리

!Tips & Links

※ js함수 선언하는 6가지 방법 <https://dmitripavlutin.com/6-ways-to-declare-javascript-functions/>

①	Function Declare (start with function) 함수 선언	once declare => invoke it in many places 스코프 안에서 사용하는 매개변수 // 이벤트핸들, 재귀recursion에 용이	재귀함수 Recursion function function factorial(n){ if(n === 0){ return 1; } return n * factorial(n - 1); }	
②	Function Expression 함수 표현식	(function keyword is present somewhere in the middle of the statement) 변수에 object로 지정 / object에 method 만들 / callback function if문 등 조건 내 함수선언declare 시, <u>ReferenceError: ~ is not defined</u> (expression으로 let ~, ~=function(){~~} 사용해야)		
②	Named Function Expression 이름있는 함수 표현식	함수를 변수에 assign 익명함수 추론 함수 내에서 이름 사용 함수 밖에서 변수.name확인	const getType = function funName(val){ return typeof val; } getType(3) = number getType.name <= funName typeof funName <= undefined	
③	Shorthand Method Definition 메소드 축약 정의	ES2015 class에 사용, function name으로 define shorter syntax => easier to understand	const collection = { item:[], add(items){ - }, get(idx){ - } } collection.add(~~) collection.get(-)	class Star { constructor(name){ this.name = name; } getMessage(msg){ return this.name+msg; } } const sun = new Star("Sun") sun.getMessage() is shining' => 'Sun is shining'
④	Arrow Function 화살표 함수	create short callbacks const absVal = (n) => { n<0?return -n:""; retrun n; } // absVal(-10) => 10		
⑤	Generator Function* 생성자 함수	함수 끝에 asterisk(*)붙임 / return a Generator object / 비동기처리 yield <~> next(); 왔다리 갔다리		
⑥	New Function (특정 상황에서 사용)	closures cannot be created=> always global // NodeJS 글로벌 오브젝트 쓸 때 const globla = new function(val1, val2, 'return val1+val2')		

/* -----<lang: en>----- */
succint [sə(k)'siŋg(k)t]: 간결 / factorial [faktôrēəl]: 계승 / mechanism <-스펠링
inferred추론 / delimiter [dē'limit] : 구분하다