Pytania



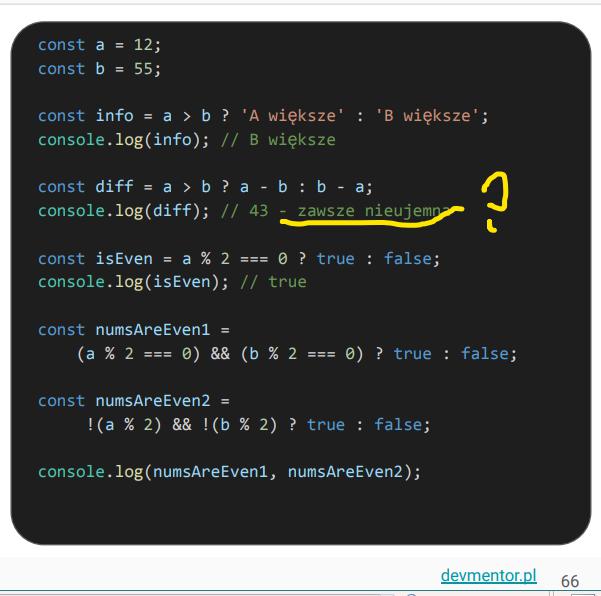
czwartek, 24 grudnia 2020 19:13

1) Cytat: Materiały Str 21

"Spotkasz się z nim w starszych skryptach lub takich, które muszą wspierać starsze przeglądarki."

Czy to znaczy że w starszych przeladarkach trzeba użyc var, czy to jest jakas implementacja wsteczna I to zadziała samoistnie?

1. O co Ci chodzi z tym zapisem? Po prostu zawsze większa, tzn przedział dodatni wartość bezwzględna większa, przedział ujemny wartość bewzgledna mniejsza



1. Specyfika wykonania pętli switch w porownaniu do if else Teoretycznie switch jest korzysniejsza dla pamiec

Bo nie jest sprawdzany kazdy if po kolei tylko jest brany konkretny case Coś takiego było ale nie wiem czy czego nie pomieszałem



\*4) Czy odwrócenie inkremtacji nam cos da ? Już wiem, że nic nam nie da. Nawet już orientuje się mniej wiecej dlaczego tak się dzieje, trochę czytałem, trochę mówiła mi Ewa – ale jakoś jeszcze mam wątpliwość xD

1. Filmik ze str 107 materialy M4 jest niedostepny ;)
2. Czy zapisay



Const arr = new Array(0);

Const arr = new Array();

Rożnią się czymś ? W konsoli tego nie widzie? Musimy pisać to 0

Przed czymś nas to chroni?

7)Jaka jest ogólna nazwa na te funkcjie które nie potrzebują operatora . z obiektem odwoławczym? Gdzieś to było ale nie mogę tego znaleźć, były funkcje wyższego rzędu ale to nie o chyba np = 2.332;

parseInt(numberX)- to jakaś grupa funkcji .forEach(numberList) - to jakaś inna grupa funkcji bo trzba użyć operatora . (a właśnie jak tutaj nazwiemy ten operator? Bo normalnie "kropka" to operator notacyjny, ale tutaj to …. hmmm – dobra bo już może miesza mi się )

1. Na ile w tym momęcie funkcjonuje I powinniśmy zastanawiać sie nad ES7 – ale to może pytanie po module 8
2. Specjalne znaki w języku JavaScript - zastosowanie/użyteczność

praktyczna – masz coś fajnego do poczytania ?

10)

Z czego wynika to że

Slow------------------------------Mid------------------------Speed---------------> .querySelector() .getElementById() .getElemetByClassName()

v= 1 v≈2,…. v= ? tak mniej więcej o ile szybsze to jest ?

Zdawałoby się że przez id będzie szybciej – jednoznacznie + są przecież zmienne z id – HTML

11)

//?dev? Z czego wynika ta różnica w wyniki, tzn. co tak naprawdę kryje się pod +(zmienna)?

const pointNumber = ".01 dasd";

console.log(parseFloat(pointNumber)); // 0.01

console.log(+pointNumber);//NaN - bezpieczna konwencja +(pointNr)

* + bo pomija wartości zawierające znaki inne niż number

1. Czy analogiczna/jakaś różnica istneje pomiędzy efektem zastosowania operatora rozproszonego

[...NameList] vs. Array.from() ?

1. obiekty tablicopodobne – forEach możemy użyć na NodeList, ale jeśli chcemy zastosować inne metody tablicowe o musimy zamienić ten objekt na objecy Array.

// muszę to jeszcze sprawdzić, ale tak mi się wydawało jak robiłem zadania I sie zastanawiałem dlaczego tak jest skoro forEach jest z proto Array -

\* Teraz jak to pisałem to widze że NodeList ma

///vv\_\_\_\_a o tym to sobie Marek poczytaj jeszcze\_\_\_vv

[entries()](https://developer.mozilla.org/en-US/docs/Web/API/NodeList/entries)

[forEach()](https://developer.mozilla.org/en-US/docs/Web/API/NodeList/forEach)

[item()](https://developer.mozilla.org/en-US/docs/Web/API/NodeList/item)

[keys()](https://developer.mozilla.org/en-US/docs/Web/API/NodeList/keys)

[values()](https://developer.mozilla.org/en-US/docs/Web/API/NodeList/values)



Ale tak dla mojej świadomoście, bo czuje że nie do końca to rozumiem

NodeList.forEach() vs. Array.forEach()

To te same metody, ale one są zdublowane I w zależności której używamy to odwołujemy się wówczas do innego proto czy może jest tak że NodeList.forEach() jest zrobione na Array.forEach() I dlatego proto arr działa na proto NodeList więc działą na NodeList?

Kurde nie kumam, bo wówczas też pozostałe funkcje by działały a przecież nie działą. Wiec na pewno tak nie jest... zatem jak jest Mati? ;D

* + znów pojawił mi sie tu temat Polyfill z poprzedniego pdf'a z pytanimi

1. To jest fragment z Operacje na atrybutach

Dlaczgo pojawia się tutaj to undefined?





\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mały powrót do RWD

1. W takim skrócie, hasłowo jest może gdzieś jakiś algorytm postępowania dla RWD? Mógłbyś coś doradzić?

np. 1) spr urzadzenia 2) spr przegladarki 3) ustal breakpoints 4) zacznij od first mobile (I teraz zaczynają się schody I o ten etap postępowania mi chodzi. Jak Ty to robisz?)

1. Ustal wysokość sekcji poprzez Xvh I a szerokość poprzez 100%
2. rozmiary el. Potomnych sekcji określaj ….

Czym najwygodnie ? Zawsze się zastanawiam I potem to chrzanie, a wolałbym na ten moment przyjąć jedną strategie I się jej trzymać.

* chodzi mi o to że jeśli używamy np wartości w procentach to gdzie wyznaczyć nasz stały punkt odniesienia żeby potem nie było niespodzianek,

7) a czcionka... jak postępujesz z czcionką?

btw8) Korzystasz ze scale? Czy lepiej tego nie ruszać xD ?

15) reset.css vs. normalize.css

Jeżeli poniższe definicje dobrze oddają sens w/w to:

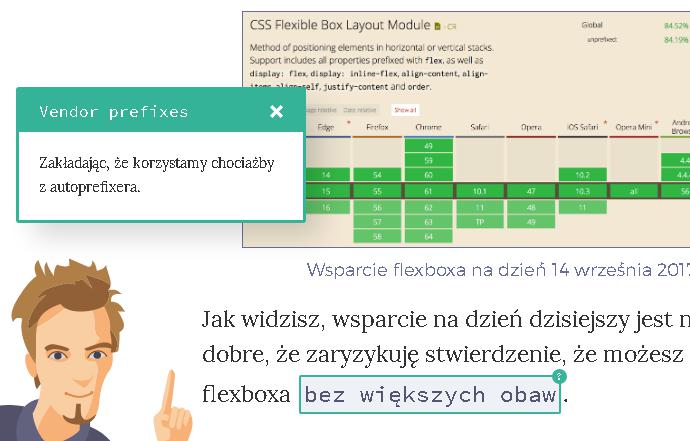
Resetowanie – ma na celu usunięcie wszystkich domyślnych stylów przeglądarki. Na przykład marginesy, wypełnienia, rozmiary czcionek wszystkich elementów są resetowane.

Normalizowanie – zachowuje przydatne style domyślne zamiast cofania wszystkiego. **Poprawia również** **różnice w wyświetlaniu elementów w różnych przeglądarkach.**

Jaką strategie lepiej obrać? Czyżby lepiej korzytać normalize?

Co się kryje pod tym poprawia wyświetlenie?

1. Mógłbyś zasugerować jakieś przystępne w formie źródlo odnośnie tych prefiksów dla flexbox? + 16.a) Plus o tych polifilach



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

17) Kojarzysz może taką wtyczkę ?

