

Lesson Plan - 13

Name of Lesson

Summary

1. DRY, KISS, YAGNI
2. Bo'lim tag lari (*header, main, section, article, aside, footer...*)
3. Background-image
4. Position (static, relative, absolute, fixed, sticky)

Dry(*Don't Repeat Yourself*) - o'z o'zingni takrorlama!

Bu sizda takroriy kod bo'lmasligi kerak degan ma'noni anglatadi. Faqat bitta joyda bo'lgan kodni boshqarish osonroq, chunki koddagi biror narsani o'zgartirish kerak bo'lsa, faqat bitta joydan o'zgartirish oson bo'ladi. Bundan tashqari, agar siz ikki yoki undan ortiq joyda bir xil kodga ega bo'lsangiz, dasturingizni yozish davomida ushbu kodni o'zgarish ehtimoli yuqori va o'zgartirish kk bo'lganda, siz bir xil kodni bitta joydan emas bir nechta joydan o'chirishingizga to'g'ri keladi. Ikki nusxadagi kod, shuningdek, kodni murakkabroq va keraksiz ko'proq qiladi.

Bundan tashqari,

- noaniq kod yozmasligingiz kerak.
- Sizning *classlaringiz*, o'zgaruvchilaringiz(*variables*), *funktsiyalaringiz*, ular ma'lum bir nomga ega bo'lishi kerak va ularning nomi ularning nimaga javob berishiga mos kelishi kerak
- Agar sizda funktsiya mavjud bo'lsa, undagi kodni o'qishga hojat qoldirmasdan, uning nomini o'qib, funktsiya nima qilishini bilishingiz kerak.

- BIR XIL VAZIFANI BAJARARUVCHI 2XIL CODE YOZMASLIK KERAK GARCHI 2LA KODINGIZNI ICHIDA YOZILGAN QIYMATLAR BOSHQACHA TARZDA YOZILGAN BOLSAYU LEKIN BIR XIL NATIJA BERAYOTGAN BOLSA XAM BU YO'L XATO


KISS(Keep It Simple, Stupid) - *oddiy saqlang ****!*

Bu prinsip sizning kodingizni soddalashtirish haqida gapiradi. Keraksiz murakkablikdan qochishingiz kerak. **Oddiy kod, uni saqlash va tushunish osonroq.**

Ushbu tamoyilni - loyihalashda va kodni yozish jarayonida qo'llashingiz mumkin. Siz

- takrorlangan kodni o'chirishingiz,
- keraksiz xususiyatlarni(features) olib tashlashingiz,
- keraksiz o'zgaruvchilar(variables) va methodlardan foydalanmaslingiz,
- o'zgaruvchilar va methodlar uchun mantiqiy va ularning mas'uliyatiga mos keladigan nomlardan foydalanishingiz
- har doim iloji bo'lsa, dasturlash davomida kod yozishning ma'lum standartlariga amal qilishingiz .

Ba'zan siz kod yozish davomida yangi ozgarishlarni amalga oshirishingiz kerak bo'lsa, siz shunchaki foydalanayotgan dasturlash tilining xususiyatlaridan(features) foydalanishingiz mumkin. Buning uchun siz ishlayotgan dasturlash tilining xususiyatlarini bilsangiz bo'lganii.



Agar siz allaqachon deploy bo'lgan, yani produtionga chiqqan saytda qandaydir o'zgartirishlar qilayotgan bo'lsangiz va u kerak bo'lmagan yoki oddiyroq bo'lishi mumkin bo'lgan narsani ko'rsangiz, uni qayta o'zgartirishingiz mumkin(refactoring).

YAGNI(You Ain't Gonna Need) - *Sizga kerak emas!*

YAGNI qisqartmasi "Sizga kerak emas" degan ma'noni anglatadi. Bu Extreme Programming (XP) dasturiy ta'minotini ishlab chiqish metodologiyasining printsidir. Ushbu tamoyil sizga kerak bo'lmagan xususiyatlarni yaratmaslik kerakligini aytadi.

Bu printsip KISS printsipiga o'xshaydi, chunki ikkalasi ham oddiyroq yechimga intiladi. Ularning orasidagi farq shundaki, YAGNI keraksiz funksionallik va mantiqni olib tashlashga, KISS esa murakkablikka e'tibor qaratadi.

Bu shuni anglatadiki, siz biror kun sizga kerak bo'lishi mumkin deb o'ylaganingiz uchun funksionallikni amalga oshirmasligingiz kerak, balki uni haqiqatan ham kerak bo'lganda amalga oshirishingiz kerak. Shunday qilib, siz hatto kerak bo'lmagan va hech qachon foydalanilmaydigan ilovalar bilan vaqt sarflashdan qochasiz.

Xulosa

Ushbu tamoyillarga rioya qiling, bu sizga yaxshiroq kod yozish imkonini beradi. Esda tutingki, toza kod (clean code) - tushunish osonroq va u biror narsani o'zgartirish yoki amalga oshirish kerak bo'lganda vaqtingizni tejaydi. Kodni takrorlashdan saqlaning, kodingizni iloji boricha sodda saqlashga harakat qiling va haqiqatan ham zarur bo'lganda funksiyalarni qo'llang.

Semantic tags

Semantik elementlar = ma'noga ega elementlar.

Semantik element brauzer va dasturchilari uchun tagning ma'nosini aniq tasvirlab beradi.

Semantik bo'lmagan elementlarga misollar: <div> va - uning mazmuni ya'ni unda qanday content borligi haqida hech narsa aytmaydi.

Semantik elementlarga misollar: <header>, <table> va <article> - uning mazmunini aniq belgilaydi.

Ko'pgina veb-saytlarda **navigatsiya**, **sarlavha** va **footer** ko'rsatish uchun quyidagi HTML kodlari mavjud:

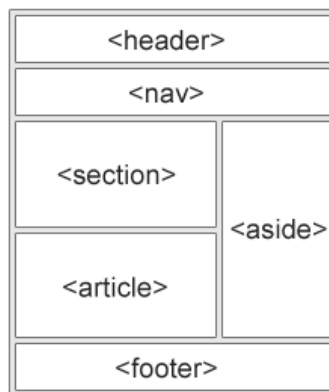
<div id="nav">

<div class="header">

<div id="footer">.

HTMLda veb-sahifaning turli qismlarini aniqlash uchun ishlatilishi mumkin bo'lgan ba'zi semantik elementlar mavjud:

- `<article>`
- `<aside>`
- `<details>`
- `<figcaption>`
- `<figure>`
- `<footer>`
- `<header>`
- `<main>`
- `<mark>`
- `<nav>`
- `<section>`
- `<summary>`
- `<time>`



`<article>` - semantic tegini qanday xolatlarda ishlatamiz?

- Forum xabarlari
- Blog postlari
- Foydalanuvchi sharhlari(comment)
- Mahsulot kartalari(Product Cards)
- Gazeta maqolalari

`<section>`

- Boblar
- Kirish (Introduction)
- Yangiliklar
- Bog'lanish uchun ma'lumot (Contact Information)

<header>

- kirish content yoki navigatsiya havolalari to'plami uchun container vazifasini o'taydi.
- Bitta yoki ko'proq heading elements (<h1> - <h6>)
- logo yoki iconkalarni o'rab turish
- authorship information(Muallif informatsiyasi)

Eslatma: Siz bir nechta <header> elementlarini bitta HTML documentida ishlatishingiz mumkin. Biroq, <header> elementini <footer>, <address> yoki boshqa bir <header> elementini ichida yoza olmasiz.

<footer>

Ko'pincha HTML documentida eng pastda ishlatiladi

<footer> elementi hujjat yoki bo'lim uchun pastki ko'rsatkichni belgilaydi.

<footer> elementi odatda quyidagilarni o'z ichiga oladi:

- mualliflik ma'lumotlari
- mualliflik huquqi haqida ma'lumot(copyright information)
- Bog'lanish uchun ma'lumot
- sayt xaritasi
- yuqori havolalarga qaytish
- tegishli hujjatlar

Bitta hujjatda bir nechta <footer> elementlar bo'lishi mumkin.

<nav>

<nav> elementi navigatsiya havolalari to'plamini belgilaydi.

Eslatma:

hujjatning barcha havolalari(links) <nav> elementi ichida bo'lishi kerak emas. <nav> elementi faqat navigatsiya havolalarini o'rab turish uchun mo'ljallangan.

Brauzerlar, masalan, nogiron foydalanuvchilar uchun [ekranni o'qiydiganlar](#), ushbu elementdan ushbu kontentning dastlabki ko'rsatilishi kerak bolgan linklarni korsatish uchun foydalanishi mumkin.

[Ekranni o'qiydiganlar](#) : Nogironligi bor shaxslar(ko'zi ochiz)lar uchun veb-sahifani to'liqligicha o'qib beradigan robot bor.

<aside>

elementi o'zi joylashtirilgan contentdan tashqari ba'zi contentni belgilaydi (masalan, yon panel, sidebar, toolbar, navbar).

<aside> kontent atrofdagi tarkib bilan bilvosita bog'liq bo'lishi kerak.

<figure>

Elementi rasmlar, diagrammalar, fotosuratlar, kodlar ro'yxati va boshqalar kabi o'z-o'zidan tashkil topgan content(self-contained content)ni o'rab turish uchun ishlatiladi

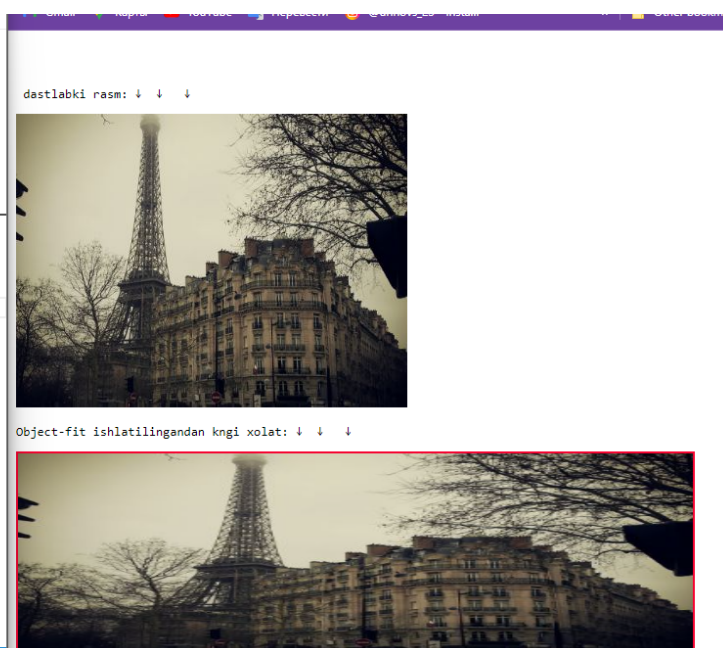
<figcaption> tegi <figure> elementi uchun sarlavhani belgilaydi. <figcaption> elementi <figure> elementining birinchi yoki oxirgi bolasi sifatida joylashtirilishi mumkin.

[LINkni bosing](#)

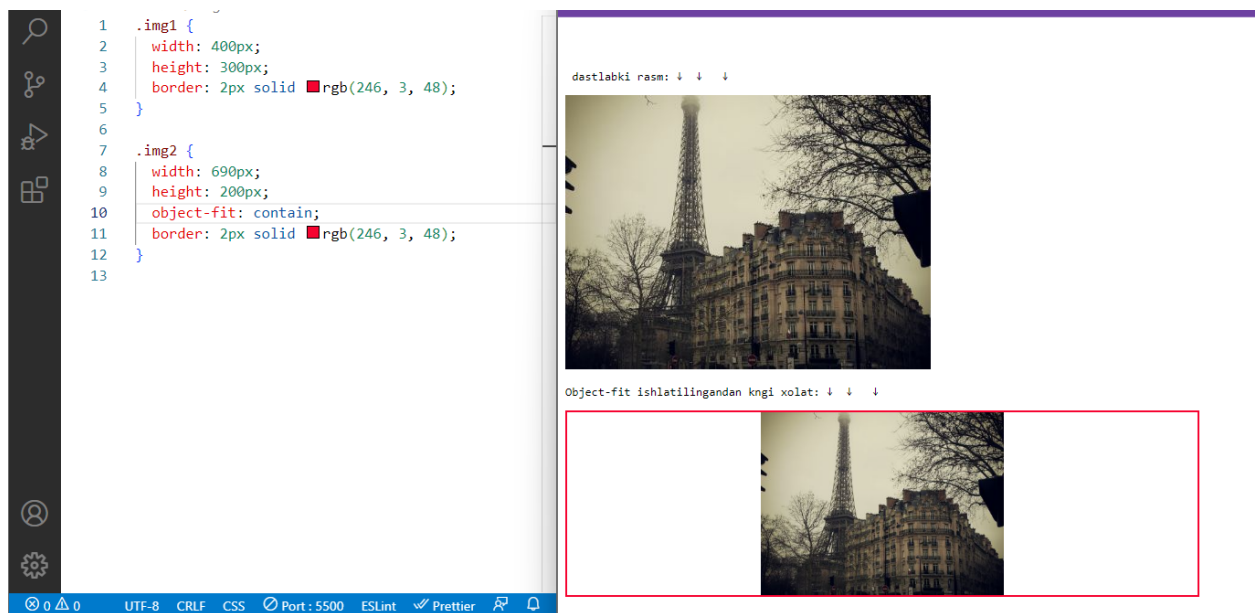
Object-fit:

fill- bu standart. Tasvirning o'lchami berilgan o'lchamni to'ldirish uchun o'zgartiriladi. Agar kerak bo'lsa, rasm mos keladigan tarzda cho'ziladi yoki siqiladi

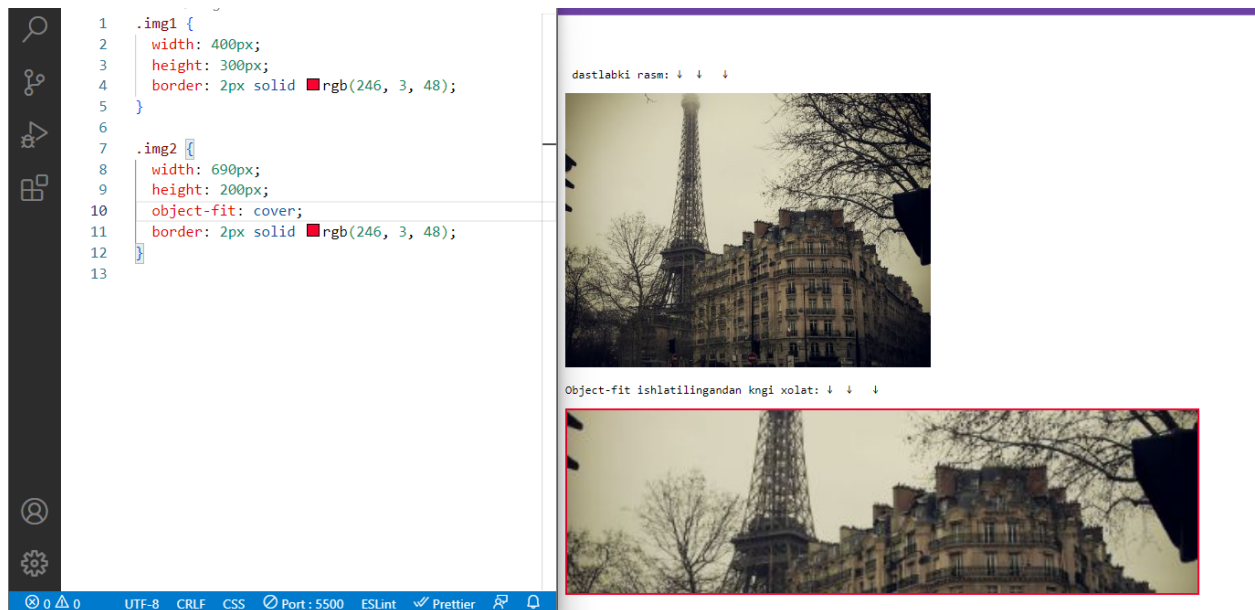
```
main.css > ...  
1 .img1 {  
2   width: 400px;  
3   height: 300px;  
4   border: 2px solid rgb(246, 3, 48);  
5 }  
6  
7 .img2 {  
8   width: 690px;  
9   height: 200px;  
10  object-fit: fill;  
11  border: 2px solid rgb(246, 3, 48);  
12 }  
13
```



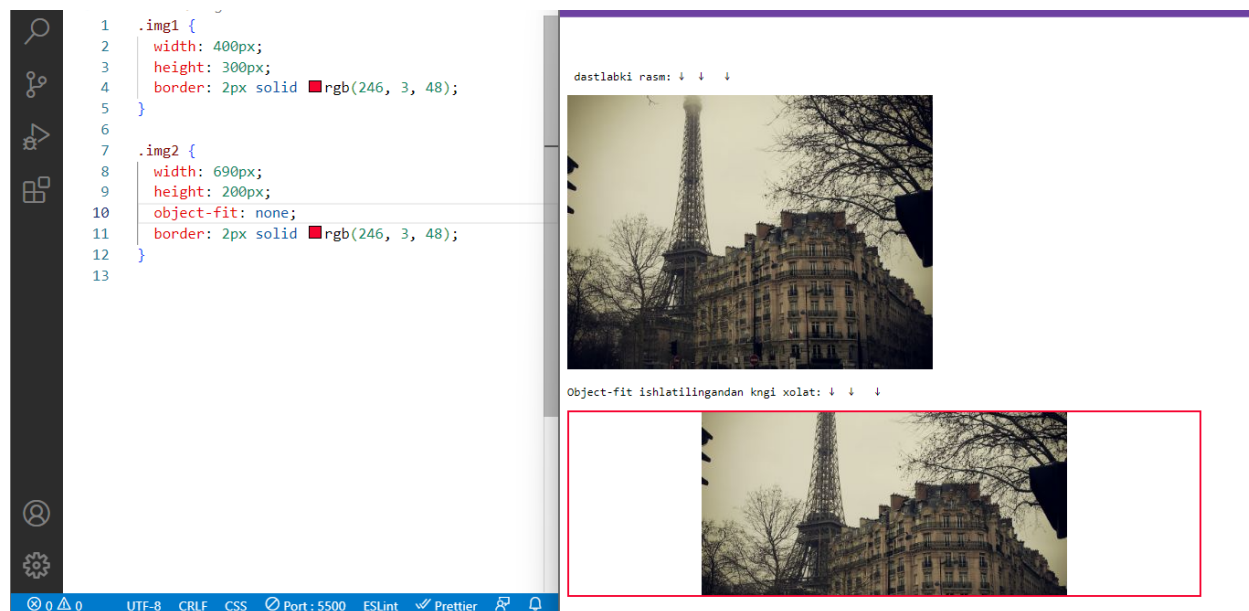
contain - Tasvir o'zining tomonlar nisbatini saqlaydi, lekin berilgan o'lchamga mos kelishi uchun o'lchami o'zgartiriladi



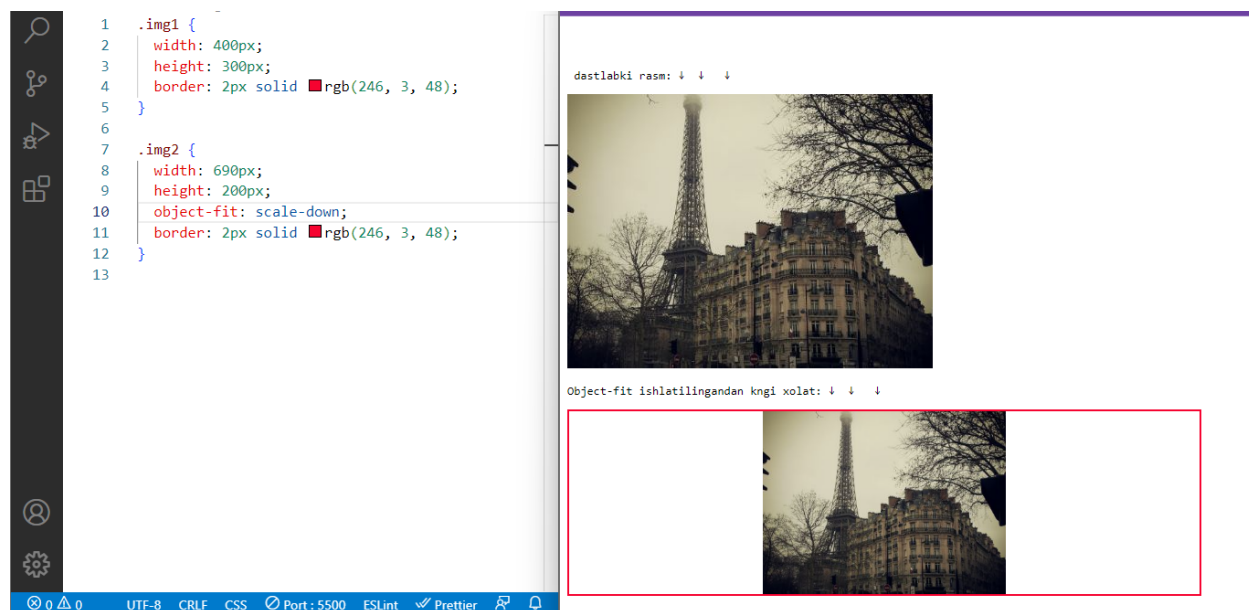
cover - Tasvir o'zining nisbatlarini saqlab qoladi va berilgan o'lchamni to'ldiradi. Tasvir mos keladigan tarzda kesiladi




none - Rasm o'lchami o'zgartirilmagan



scale-down - tasvir hech qaysi yoki tarkibida eng kichik versiyasiga kichraytiriladi





Background-image:

Bg-repeat

Bg-position

Bg-size

Position:element joylashuv turini belgilab beradi.


Static - default (top, left, right, bottom qiymatlar tasir qilmaydi)

Relative- elementning dastlabki holatiga nisbatan joylashadi.

Fixed: element viewportga nisbatan joylashadi, va xar doim element o'zining joyida qoladi, scroll paytida joyini o'zgartirmaydi (inline)

Absolute - o'ziga eng yaqin position:static bolmagan otasiga nisbatan joylashadi

Sticky - scrollni o'zgarishiga qarab joylashadi, relative va fixed qiymatlarini o'zida saqlidi va scroll qayerda joylashganiga qarab shu 2 qiymatdan birini oladi



```
div.sticky {  
  
  position: -webkit-sticky;  
  
  position: sticky;  
  
  top: 0;  
  
  padding: 5px;  
  
  background-color: #cae8ca;  
  
  border: 2px solid #4CAF50;  
  
}
```

<p>Try to scroll inside this frame to understand how sticky positioning works.</p>

<div class="sticky">I am sticky!</div>

<div style="padding-bottom:2000px">

<p>In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position.</p>

<p>Scroll back up to remove the stickiness.</p>

<p>Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.</p>

<p>Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.</p>

</div>

—