

HTML PITANJA:

1. ZADATAK: Ubaciti link za audio file-ove sa kontrolama

```
<audio controls>
  <source src="audio.mp3" type="audio/mpeg">
  <p>Vaš preglednik ne podržava reprodukciju audio sadržaja</p>
</audio>
```

2. Kojim se elementom označava uvodni sadržaj? (div, head, header ili ništa od navedenog)

```
<head></head>
```

3. Kako napisati audio komponentu s kontrolama:

```
<audio controls>
<source src="horse.ogg" type="audio/ogg">
<source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio tag.
</audio>
```

4. Može li se article koristiti više puta?

Da

5. Što se koristi za glavnu navigaciju?

nav

6. Na zadanom primjeru objasnite razliku između HTML oznake i atributa:

```
<audio controls src="glazba/jingl.mp3"></audio>
```

HTML oznaka: audio

- određuje tip HTML elementa te daje značenje njegovom sadržaju

HTML atributi: controls, src

- pruža dodatne informacije o sadržaju HTML elemenata

7. Potrebno je HTML dokumentu dodijeliti meta podatak koji opisuje njegov sadržaj u formi sljedeće rečenice “Službene stranice Učilišta Algebra”. Što je potrebno napisati unutar <head> oznake HTML dokumenta kako se to postiglo?

```
<meta name="description" content="Službene stranice Učilišta Algebra" />
```

8. Potrebno je HTML dokumentu dodijeliti oznaku jezika sadržaja “en”. Koji atribut je potrebno dodijeliti elementu <html> kako bi se to postiglo?

```
<html lang="en"></html>
```

9. Kod otvaranja HTML dokumenta u web pregledniku u njegovom zaglavlju ispisuje se naslov “Učilište Algebra”. Pomoću kojeg HTML elementa je naslov dokumenta postavljen?

```
<title>Učilište Algebra</title>
```

10. Zadan je sadržaj: <https://example.com/movie.mp4> Pomoću HTML-a objavite video sadržaj s kontrolama za upravljanje reprodukcijom sadržaja. Video formi dodijelit dimenzije 600x400px.

```
<video src="https://example.com/movie.mp4" controls width="600" height="400"></video>
```

11. Zadan je sadržaj na sljedećoj relativnoj putanji: galerija/panorama-grada.jpg Pomoću HTML-a objavite grafiku te joj dodijelite dimenzije 720x500px. Postavite zamjenski tekst “Panorama grada”.

```

```

12. Koji element koristimo za označavanje sadržaja koji predstavlja podnožje tijela stranice ili podnožje određene sekcije (sadrži autorska prava, kontakt podatke itd.)

```
<footer></footer>
```

13. Koji element koristimo za označavanje sadržaja koji predstavlja cjelinu neovisnog (samostalnog) sadržaja u dokumentu?

```
<article></article>
```

14. Koji element koristimo za grupiranje sadržaja isključivo radi stiliziranja pomoću CSS-a (bez da mu damo značenje)?

`<div></div>`

15. Koji element koristimo za tematsko grupiranje sadržaja?

`<section></section>`

16. Što predstavlja pristupačnost web rješenja= Navedite nekoliko primjera kako se ona postiže.

Pristupačnost web rješenja: Mogućnost da se web rješenje koristi prema potrebama i mogućnostima korisnika. Postiže se:

- Primjenom semantičkih HTML elemenata
- Pravilnom definicijom HTML atributa (npr. “alt” kao zamjenski tekst za grafike, “role” za davanje značenja elementima, “lang” kao oznaka jezika sadržaja itd.)
- Osiguranjem tekstualnog sadržaja kao alternative za ne-tekstualni sadržaj (npr. tekstualni transkript za audio sadržaj)
- Omogućavanjem korisnicima da prilagode vlastito sučelje (veličinu znakova, boje itd.)

17. Koji HTML element koristimo, a da se njegov sadržaj ne tiče direktno okoline?

`<aside></aside>`

CSS PITANJA:

1. ZADATAK:

Prazan `<div>` element.

Koristeći CSS; selektirajte element i uredite ga tako da odgovara obliku na slici. Pretpostavite veličine i boje, one ne moraju biti u potpunosti točne.

```
div {
    background-color: red;
    width: 200px;
    height: 200px;
    border-top-left-radius: 200px;
    border-bottom-right-radius: 200px;
}
```



2. Nadopunite isječak koda tako da elementu dodijelit crvenu boju znakova preko linijskog zapisa

```
<p>Lorem ipsum dolor</p>
```

```
<p style="color: red;">Lorem ipsum dolor</p>
```

3. Kako se poziva varijabla u CSS-u? / Kako se pozivaju var(--)? → 4 ponuđena odg.
4. Media queries
5. Smije li child mijenjati varijablu koju dobije od parenta?
6. Odabрати kako se poziva @include.
7. Nadopunite isječak koda tako da u zaglavlju HTML dokumenta dohvatite vanjsku CSS datoteku sa sljedeće lokacije: css/style.css

```
<head>
</head>
<head>
    <link rel="stylesheet" href="css/style.css" />
</head>
```

8. Nadopunite isječak koda odgovarajućim selektorom tako da dohvatite sve HTML elemente koji sadrže atribut title s točnom vrijednošću "Učilište Algebra".

```
{
    display: inline-block;
    padding: 2em;
    color: #eee;
    font-size: 20px;
}
```

```
[title="Učilište Algebra"] {
    display: inline-block;
    padding: 2em;
    color: #eee;
    font-size: 20px;
}
```

9. Nadopunite isječak koda odgovarajućim selektorom tako da dohvatite sve HTML elemente koji sadrže atribut title u čijoj vrijednosti se bilo gdje nalazi izraz “Algebra”.

<pre>{ display: inline-block; padding: 2em; color: #eee; font-size: 20px; }</pre>	<pre>[title*="Algebra"] { display: inline-block; padding: 2em; color: #eee; font-size: 20px; }</pre>
---	--

10. Nadopunite isječak koda odgovarajućim selektorom tako da dohvatite sve HTML elemente koji sadrže atribut title na početku čije vrijednosti se nalazi izraz “Algebra”

<pre>{ display: inline-block; padding: 2em; color: #eee; font-size: 20px; }</pre>	<pre>[title^="Algebra"] { display: inline-block; padding: 2em; color: #eee; font-size: 20px; }</pre>
---	--

11. Nadopunite isječak koda odgovarajućim selektorom tako da dohvatite sve posjećene poveznice <a> (one koje su već bile otvarane)

<pre>{ display: inline-block; padding: 2em; color: #eee; font-size: 20px; }</pre>	<pre>a:visited { display: inline-block; padding: 2em; color: #eee; font-size: 20px; }</pre>
---	---

12. Nadopunite isječak koda odgovarajućim selektorom tako da dohvatite sve slike iznad kojih je postavljen pokazivač miša.

<pre>{ transform: scale(1.1); }</pre>	<pre>img:hover { transform: scale(1.1); }</pre>
---	---

13. Nadopunite osječak koda odgovarajućim selektorom tako da dohvatite sve paragrafe <p> koji su prvo dijete, određenog tipa, svog roditelja.

<pre>{ color: #eee; font-size: 20px; }</pre>	<pre>p:first-of-type { color: #eee; font-size: 20px; }</pre>
--	--

14. Nadopunite isječak koda odgovarajućim selektorom tako da ugradite sadržaj “Upamti: “ sa zadanim stilovima prije postojećeg sadržaja naslova 1. Razine <h1>.

```
{
  content: "Upamti: ";
  color: red;
  background-color: yellow;
}

h1::before {
  content: "Upamti: ";
  color: red;
  background-color: yellow;
}
```

15. Vrijednost varijable “text-color” postavite kao boju znakova svih paragrafa.

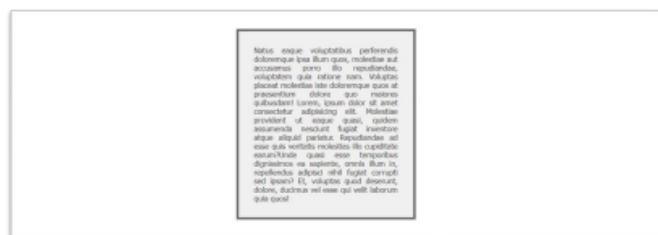
```
:root {
  --text-color: #333;
}

p {
}

:root {
  --text-color: #333;
}

p {
  color: var(--text-color);
}
```

16. Zadan je <article> element koji sadrži tekstualni sadržaj. Koristeći CSS selektirajte element i uredite ga tako da odgovara obliku na slici (pomoću margina centrirajte područje sadržaja elementa). Pretpostavite veličine i boje, one ne moraju biti u potpunosti točne.



```
article {
  background-color: #eee;
  text-align: justify;
  width: 300px;
  border: 5px solid #666;
  margin: 2em auto;
  padding: 2em;
}
```

17. Zadan je <article> element koji sadrži tekstualni sadržaj. Koristeći CSS selektirajte element i uredite ga tako da odgovara obliku na slici (pomoću margina područje sadržaja elementa postavite na desnu stranu): Pretpostavite veličine i boje, one ne moraju biti u potpunosti točne.



```
article {
  background-color: #eee;
  text-align: justify;
  width: 25%;
  border-left: 8px double #666;
  border-right: 5px dashed #666;
  margin: 2em 0 2em auto;
  padding: 2em 4em;
}
```

18. Na prazno mjesto upišite izraz za poziv SCSS funkcije “margin” te joj dodijelite dva cjelobrojna argumenta.

```
@function margin($a, $b) {
  @return $a * $b;
}

p {
  margin-top: _____ * 1px;
}
```

```
@function margin($a, $b) {
  @return $a * $b;
}

p {
  margin-top: margin(3, 2) * 1px;
}
```

19. Unutar pravila koju obuhvaća sve elemente tipa <selection> primijenite SCSS mixin “flex-container”.

```
@mixin flex-container {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
}

section {
}
```

```
@mixin flex-container {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
}

section {
  @include flex-container;
}
```

20. Dohvatite “partials” datoteku sa sljedeće lokacije: base/_color.scss

```
@import "base/color";
```

21. Kako se naziva proces prilagodbe CSS koda za produkciju, čime se smanjuje veličina datoteke te ubrzava njeno preuzimanje sa poslužitelja?

Proces se naziva minifikacija ili minimizacije CSS koda.

22. S kojim izrazom uključujemo mixin stilove?

@include

23. Kako se piše varijabla SCSS-u.

Na biranje je, možda var(--blue).

CSS varijable

- Spremnik za vrijednost koja se može više puta iskoristiti unutar dokumenta
 - naziv joj započinje sa znakom "--"
 - pristupa joj se preko funkcije var()

```
:root {
  --blue: #1e90ff;
  --white: #ffffff;
}

body {
  background-color: var(--blue);
}
```

24. Kako ubaciti inline CSS?

Linijski CSS

- unutar HTML oznaka → dodavanje atributa “style” HTML elementu
 - ne koriste se selektori
 - ograničava primjenu CSS pravila samo na element u kojemu su napisana
 - daje nepregledan kod
- primjer:
 - `<p style="color:red;font-size:16px;">Odlomak teksta</p>`

25. Dohvatite “partials” datoteku sa sljedeće lokacije: `base/_color.scss`

`@import "base/color"`

26. Kako bi koristio varijablu s bojom u CSS-u?

```
:root {  
    --main-bg-color: #c1c1c1;  
}  
  
body {  
    background-color: var(--main-bg-color);  
}
```

JAVASCRIPT PITANJA:

1. Pravilna sintaksa za pozvati funkciju.
2. ECMAScript /Što je ECMA?

JavaScript jezik razvija se prema ECMAScript specifikaciji:

- opisuje sintaksu i ponašanje programskog jezika
- definira standard za implementaciju JavaScript-a

3. Koja je razlika između *let* i *const* deklaracije varijabli.

Const izraz koristimo i definiciji spremnika za podatke ako želimo onemogućiti promjenu primitivne vrijednosti - vrijednost varijable ne može se mijenjati.

Let deklaracija označuje da planiramo mijenjati varijablu, a opseg varijable je ograničen unutar bloka, a ne funkcije

4. Unutar petlji možemo koristiti izraze “continue” i “break”. Koja je razlika između njih?

Izraz “continue” prekida trenutnu iteraciju (odnosno trenutno izvršavanje bloka naredbi) te nastavlja s idućom. Izraz “break” prekida izvršavanje petlje te izlazi iz nje.

5. BrowserObjectModel (BOM)

Skupina unaprijed definiranih objekata koji stranici (JavaScript-u) omogućuju interakciju s web preglednikom.

Iako ne postoji službeni standard, većina web preglednika ipak ima implementirano identično sučelje za komunikaciju s JavaScript jezikom.

6. HTML DOM

Platforma i jezično neutralno sučelje koje omogućuje programima i skriptama dinamički pristup i ažuriranje sadržaja, strukture i stila dokumenta.

7. Što definira DOM?

- HTML elemente kao objekte
- svojstva svih HTML elemenata
- metode pristupa svim HTML elementima
- događaje za sve HTML elemente

8. JavaScript jezik predviđen je za objektno-orijentirano ili funkcijsko programiranje?

JavaScript je skriptni programski jezik visoke razine koji je prvenstveno predviđen za objektno-orijentirano programiranje, ali se može koristiti i za funkcijsko programiranje.

9. Što od navedenog nije ispravan identifikator varijable? a) number1 **b) number-1** c) \$number1 d) number_1

10. ZADATAK:

Potrebno je definirati funkciju “max” koja će vratiti veći od dva broja. Ako su brojevi jednaki, funkcija vraća prvi broj.

Funkcija prima dva broja preko parametra. Pretpostavite da će parametri uvijek biti brojevi.

Primjer:

max(1,2) - funkcija vraća 2

max (1,1) - funkcija vraća 1

script.js

```
function max(a,b) {
    if (a>b)
        { console.log(a) }
    else if (b>a)
        { console.log(b) }
    else
        { console.log(a) }
```

```
function compare(x, y) {
    if (x > y) {
        return x;
    } else if (y > x) {
        return y;
    }
    if (x === y) {
        return x;
    }
}

console.log(compare(5, 7));
```

11. Koja metoda nije ispravna? while-do, do-while, for i još jedna
12. Koja od navedenih metoda objekta *document* nije ispravna?
 - a. document.getElementById()
 - b. document.querySelector()
 - c. document.querySelectorAll()
 - d. document.getElementByClassName()

```
document.getElementById()
document.getElementsByTagName()
document.getElementsByClassName()
document.getElementsByName()
document.querySelector()
document.querySelectorAll()
```

13. ZADATAK:

U footeru je element s određenim id-jem. Preko JS-a dodaj u taj element trenutnu godinu

```
const setDate = document.getElementById("copyright-year")  
  
setDate.innerText = new Date().getFullYear()
```

14. Zaokruživanje

Što od navedenog nije ispravan identifikator varijable?

- a) 1userData
- b) \$userData
- c) user_data
- d) user1Data

Imenovanje

Identifikatori

Identifikator je jednostavno ime za imenovanje varijabli i funkcija i za pružanje oznaka za određene petlje u JavaScript kodu. Ima par pravila kod imenovanja:

1. JavaScript razlikuje velika i mala slova
2. Naziv mora početi sa **slovom, podvlakom (_)** ili **znakom dolara (\$)**
3. Sljedeći znakovi mogu biti **slova, znamenke, podvlake** ili **\$**
4. Ključne riječi se ne smiju koristiti kao identifikatori

15. Opišite pojedine djelokruge varijabli.

Globalni djelokrug (globalne varijable)

- definiraju se izvan funkcija
- mogu se koristiti u čitavom programu, zatvaranjem stranice brišu se

Djelokrug funkcije (lokalne varijable)

- definiraju se unutar funkcije te se mogu koristiti samo unutar nje
- izlaskom iz funkcije brišu se

Djelokrug bloka

- definiraju se unutar bloka { } pomoću ključne riječi *let*
- dostupne su samo unutar bloka u kojem su definirane
- moraju se deklarirati prije upotrebe te se ne mogu ponovno deklarirati

16. Deklarirajte i inicijalizirajte varijable firstName i lastName i spojite ih u varijablu fullName, ali s razmakom (" ") između imena i prezimena.

```

var firstName = "John";
var lastName = "Doe";
var fullName = firstName + " " + lastName;
/* ili */
var fullName = firstName.concat(" ", lastName);

```

17. Deklarirajte varijable y i z te im dodijelite proizvoljne vrijednosti. Nakon toga razlomite ovu ternarnu operaciju an 3 različite operacije (upotrebom grananja). Ispišite dobivenu vrijednost varijable x.

```
var x = z === 2 ? y : 5;
```

```

var x;
var y = 10, z = 2;
if (z === 2) {
    x = y;
}
else {
    x = 5;
}

```

18. Koji izraz možemo koristiti kao alternativu “if” izraza za postavljanje višestrukog grananja?

Izraz “switch”

19. Koliko puta će se izvršiti sljedeći blok naredbi?

```

for (let count = 0; count <= 10; count+=2) {
    console.log(count);
}

```

6 puta

20. Koristeći JavaScript dohvatite HTML element preko njegova ID-a “zaglavlje” te mu dodajte novi podelement tipa <h1> s tekstom “Naslov” (novi element biti će posljednji podelement svog roditelja).

```

var roditelj = document.getElementById("zaglavlje");
roditelj.innerHTML += "<h1>Naslov</h1>";
/* ili */
var roditelj = document.getElementById("zaglavlje");
var naslov = document.createElement("h1");
naslov.textContent = "Naslov";
roditelj.appendChild(naslov);

```

21. Koristeći JavaScript dohvati HTML element preko njegova ID-a “zaglavlje” te ga uklonite iz HTML stabla.

```
var element = document.getElementById("zaglavlje");
element.remove();
/* ili */
var element = document.getElementById("zaglavlje");
element.parentNode.removeChild(element);
```

22. Koristeći JavaScript dohvatite HTML element preko njegova ID-a “zaglavlje” te mu postavite sljedeću pozadinsku boju: “#ddd”.

```
var element = document.getElementById("zaglavlje");
element.style.backgroundColor = "#ddd";
/* ili */
var element = document.getElementById("zaglavlje");
element.style.cssText = "background-color: #ddd;";
```

22. Navedene su 4 petlje, jedna je od njih kriva?

Petlje

while, do ... while

```
var count = 0;
while (count < 10) {
    console.log(count);
    count++;
}
```

```
var count = 0;
do {
    console.log(count);
    count++;
} while (count < 10);
```

for, for ... in

```
for (var count = 0; count < 10; count++) {
    console.log(count);
}
```

```
for (var i in nazivObjekta) {
    /* ... */
}
```

23. Navedeno je nekoliko getElementById, getElementsByClassName itd.. jedan ima sintaktičku pogrešku (npr. negdje je slovo veliko, a treba biti malo ili je neka druga greška. Treba izabrati krivo napisani)

Selektiranje DOM elementa

Za mijenjanje DOMa ili elementa u DOMu prvo ga moramo 'pronaći' u dokument strukturi i selektirati.

DOM definira brojne **načine odabira elemenata**, koristeći:

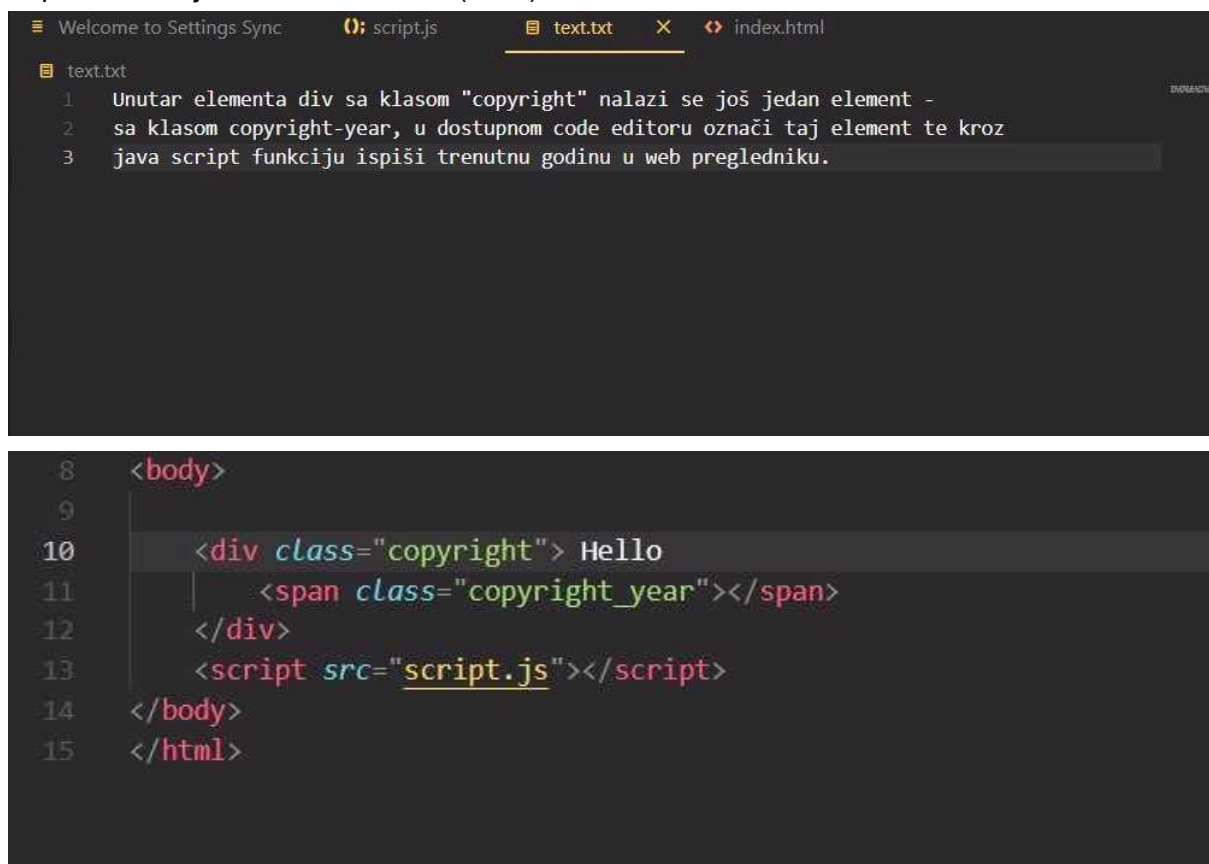
- **id elementa**
- **tag elementa (tip)**
- **CSS klasu ili drugi CSS selektor**
- atribut name

```
document.getElementById("header-title");
document.getElementsByTagName("li");
document.getElementsByClassName("description");
document.getElementsByName("gender");
document.querySelector("#header-title");
document.querySelectorAll(".description");
```

24. Kako bismo zapisali da se mijenja selektor a u trenutku prijelaza mišem?
a:hover

25. ZADATAK

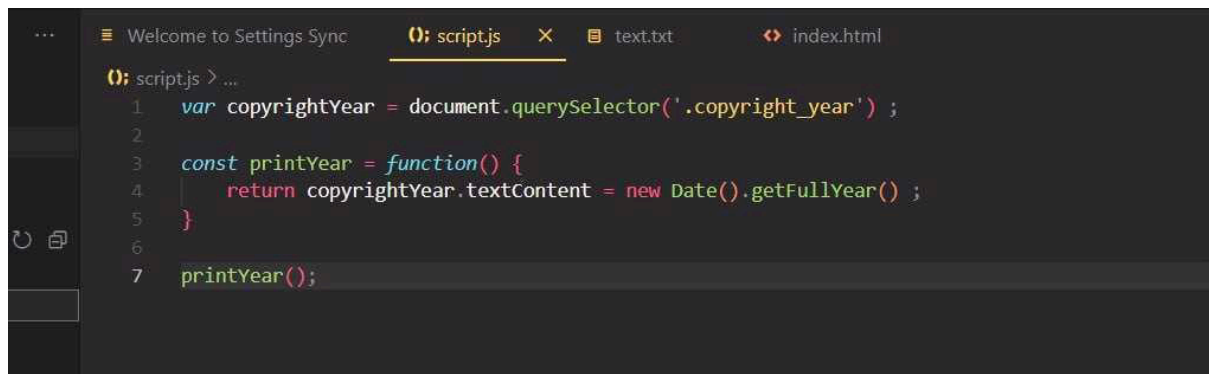
Napisati funkciju vezanu uz datum (Date):



The screenshot shows a code editor with two tabs: 'text.txt' and 'index.html'. The 'text.txt' tab is active and contains a task description in three lines. The 'index.html' tab is also visible and contains an HTML snippet. The task description asks to write a JavaScript function that prints the current year using the Date object, and to mark a specific element in the provided HTML with a class 'copyright_year'.

```
1 Unutar elementa div sa klasom "copyright" nalazi se još jedan element -
2 sa klasom copyright-year, u dostupnom code editoru označi taj element te kroz
3 java script funkciju ispiši trenutnu godinu u web pregledniku.
```

```
8 <body>
9
10 <div class="copyright"> Hello
11   <span class="copyright_year"></span>
12 </div>
13 <script src="script.js"></script>
14 </body>
15 </html>
```

A screenshot of a code editor with a dark theme. The top bar shows three tabs: 'Welcome to Settings Sync', 'script.js' (active), 'text.txt', and 'index.html'. The 'script.js' tab is selected, and the code editor displays the following JavaScript code:

```
0: script.js > ...
1  var copyrightYear = document.querySelector('.copyright_year') ;
2
3  const printYear = function() {
4      return copyrightYear.textContent = new Date().getFullYear() ;
5  }
6
7  printYear();
```

RAZVOJNA OKOLINA PITANJA:

1. Što je sustav za verzioniranje koda te zašto se koristi u web razvoju?

Alat omogućuje čuvanje i praćenje promjena u datoteci ili skupu datoteka tijekom vremena (tko je napravio promjenu i kada). Omogućuje lak oporavak u slučaju da je nešto pošlo po krivu (vraćanje datoteka u prethodno stanje) te olakšava suradnju više od jedne osobe koje rade na istom projektu.

2. Koji tipovi sustava za verzioniranje koda postoje?

- lokalni sustav za verzioniranje koda
- centralizirani sustav za verzioniranje koda
- distribuirani sustav za verzioniranje koda

3. Što je Git?

Git je distribuirani sustav za verzioniranje koda (DVCS). Omogućuje praćenje promjena u izvornom kodu tijekom razvoja softvera te koordiniranje rada među programerima.

4. Što je repozitorij?

Repozitorij je mjesto na kojem se pohranjuju sve datoteke jednog projekta. Može biti lokalni ili na nekom poslužitelju, dok na jednom repozitoriju može raditi jedna ili više osoba.

5. Napišite Git naredbu s kojom se inicijalizira Git repozitorij.

`git init`

6. Napišite Git naredbu s kojom se lokalno sprema trenutna verzija repozitorija.

`git commit -m "message"`

7. Napišite Git naredbu za zapisivanje svih lokalnih commitova na udaljeni repozitorij (npr. GitHub).

`git push`

8. Napišite Git naredbu za povlačenje promjena s udaljenog na lokalni repozitorij.

`git pull`

9. Napišite Git naredbu za zapisivanje svih lokalnih commitova na udaljeni repozitorij.

`git push`

10. Napišite Git naredbu za spajanje dvije Git grane kako bi se promjene iz jedne zapisale u drugu (u glavnu granu repozitorija - master).

`git merge`

11. Što predstavlja jQuery?

Brzu JavaScript biblioteku bogatu mnogim mogućnostima.

Značajno pojednostavljuje izvršavanje sljedećih radnji:

- upravljanje HTML stablom (DOM)
- upravljanje CSS stilovima
- primjenu efekata i animacija u radu web stranica
- upotrebu događaja za pokretanje definiranih operacija
- asinkronu komunikaciju klijenta i poslužitelja (AJAX)

12. Što je npm (node package manager)?

Snažan upravitelj programskim paketima:

- softverski alata koji automatizira proces instaliranja, nadogradnje, konfiguriranja i brisanja računalnih programa ili biblioteka (paketa)
- dolazi s instalacijom Node.js okruženja

13. Napišite npm naredbu za instalaciju paketa *lodash* i editiranje package.json datoteke (ovisnost se zapisuje u datoteku package.json).

```
npm install --save lodash
```

ili

```
npm install lodash
```


14. Napišite npm naredbu za instalaciju paketa *jest* koji se koristi samo prilikom razvoja aplikacije. Ovisnost se zapisuje u datoteku `package.json` (naredba editira `package.json` datoteku).

```
npm install --save-dev jest
```

15. Što su task runner-i?

Task runner je alat koji omogućuje automatizaciju ponavljajućih zadataka koji se tipično rade ručno tijekom razvoja projekta. Primjeri: gulp, grunt, broccoli, brunch...

16. Što je module bundler?

Alat koji se koristi prilikom izgradnje aplikacije za produkciju. Omogućuje povezivanje i optimizaciju više modula (datoteka) u jedan proizvodno spreman paket. Primjer: webpack, rollup, browserify...

17. Kako bismo u Node-u inicirali algebra-library?

```
npm install algebra-library
```

18. Naredba koja se koristi za instaliranje paketa navedenih u `package.json` datoteci.

```
npm install <lokacija paketa na disku>
```

ReactJS pitanja

1. Što je React?

React je popularna JavaScript biblioteka koja služi za izradu korisničkih sučelja. Pomoću malih izoliranih dijelova koda (komponenti) stvaramo kompleksno, efikasno i fleksibilno korisničko sučelje.

2. Napišit naredbu za kreiranje React aplikacije preko CLI-a (Command Line Interface). Naziv aplikacije glasi "react-app".

```
npx create-react-app react-app
```

3. Što je potrebno napraviti kako bi se programski kod napisan u jednoj datoteci mogao koristiti u drugoj datoteci?

Kod je potrebno:

- exportati (napraviti izvoz) iz izvorne datoteke i
- importati (napraviti uvoz) u datoteci u kojoj ga želimo iskoristiti

4. Prikazana je definicija React komponente koja sadrži grešku JSX sintaksi. Objasnite u čemu je greška.

```
function MyComponent() {  
  return (  
    <div>  
      <h1>Hello!</h1>  
      <p>How are you?</p>  
    </div>  
  );  
}
```

- Ako komponenta vraća više JSX elemenata, svaki mora biti ugniježđen u jednom vršnom elementu, najčešće elementu **<div>**

← ispravno

5. Što je potrebno upisati na prazno mjesto kako bi uključili JavaScript izraz unutar JSX-a te u sadržaj koji generira ugradili zbroj konstanti x i y?

```
function MyComponent() {  
  const x = 5, y = 7;  
  return (  
    <div>  
      <h1>Hello!</h1>  
      <p>Zbroj: {x + y}</p>  
    </div>  
  );  
}
```

- Unutar JSX-a JavaScript kod je potrebno postaviti unutar vitičastih zagrada: {JavaScript code}

← ispravno

6. Ako je vrijednost konstante “user” jednaka “administrator”, komponenta vraća poruku “Hello Friend”, u suprotnom vraća poruku “Hello Stranger”. Što je potrebno upisati na prazna mjesta kako bi se postiglo zadano uvjetno iscrtavanje?

```
function MyComponent() {  
  const user = "administrator";  
  const greetings = user === "administrator" ___ <p>Hello Friend</p> ___ <p>Hello Stranger</p>;  
  return <div>{greetings}</div>;  
}
```

ILI

```
function MyComponent() {  
  const user = "administrator";  
  const greetings = user === "administrator" ? <p>Hello Friend</p> : <p>Hello Stranger</p>;  
  return <div>{greetings}</div>;  
}
```

7. Na koja dva načina je moguće kreirati komponente unutar React biblioteke?

Komponente je moguće kreirati pomoću funkcije ili klase.

8. Objasnite što su propsi (*properties*).

Propsi su način na koji komponente međusobno komuniciraju:

- omogućuju prijenos informacija iz komponente roditelj (parent) u komponentu dijete (child).

Protok podataka kroz propse uvijek je jednosmjernan, iz komponente roditelj u komponentu dijete:

- komponenta dijete ne može i ne smije mijenjati props objekt koji je primila od roditelja
9. Što je potrebno upisati na prazno mjesto kako bi komponenti “ChildComponent” proslijedili props “name” s proizvoljnom vrijednošću (npr. “Child”)?

<pre>function ParentComponent() { return (<div> <h1>Ja sam ParentComponent</h1> <ChildComponent _____ /> </div>); }</pre>	<pre>function ChildComponent(props) { return <h1>Ja sam {props.name}</h1>; }</pre>
<pre>function ParentComponent() { return (<div> <h1>Ja sam ParentComponent</h1> <ChildComponent name="Child" /> </div>); }</pre>	<pre>function ChildComponent(props) { return <h1>Ja sam {props.name}</h1>; }</pre>

10. Što predstavlja stanje komponente te što se dogodi s komponentom ako se ono promijeni?

Stanje komponente predstavlja built-in objekt koji sadrži podatke o komponenti. Ako se stanje komponente promijeni, ona se ponovno iscrtava (renderira) kako bi se prikazala promjena u stanju. Ponovnim renderiranjem komponente renderiraju se i sve njezine child komponente.

11. Putem koje metode je jedino ispravno mijenjati stanje komponente definirane klasom, kako bi osigurali da se ona ponovno iscrtava (renderira)=

Metoda setState

12. Što omogućuju hook funkcije? Navedite nekoliko primjera hook funkcija.

Hook funkcije omogućuju upotrebu stanja (state) i drugih mogućnosti unutar komponenta definiranih funkcijom (npr. simulacija životnog ciklusa komponente)

Primjeri:

- useState hook funkcija omogućuje korištenje stanja u komponentama definiranim funkcijom
- useEffect hook funkcija omogućuje oponašanje rada metoda životnog ciklusa.

13. Nadopunite definiciju komponente tako da se klikom na gumb poziva funkcija "handleButtonClick".

```
function MyComponent() {  
  const handleButtonClick = () => {  
    console.log("Button click!");  
  };  
  
  return (  
    <button _____>Klikni me!</button>  
  );  
}
```

```
function MyComponent() {  
  const handleButtonClick = () => {  
    console.log("Button click!");  
  };  
  
  return (  
    <button onClick={handleButtonClick}>Klikni me!</button>  
  );  
}
```

14. Navedite faze životnog ciklusa React komponente.

- inicijalizacija komponente (initialization)
- postavljanje na DOM (mounting)
- ažuriranje (updating)
- brisanje iz DOM-a (unmounting)

15. Što omogućuje primjena lista unutar React komponenti? Zašto elementima liste dajemo svojstvo key s jedinstvenom vrijednošću?

Liste olakšavaju definiciju sadržaja (elemenata istog tipa) koji vraća određena komponenta te omogućuju veću fleksibilnost u upravljanju njime. Svojstvo key pomaže React-u da prepozna elemente liste koji su se promijenili, dodali ili uklonili - osigurava bolji performans.

16. Navedite tri načina za stilizaciju sadržaja komponenti u React-u.

- CSS stylesheet
- inline stilovi
- CSS Modules

17. Koji tip aplikacija gradimo s React-om?

S React-om gradimo Single-Page Aplikacije:

- web-aplikacije koje se sastoje od samo jedne stranice
- stranica u interakciji s korisnikom vrši dinamičko prepisivanje trenutnog sadržaja umjesto učitavanja nove stranice

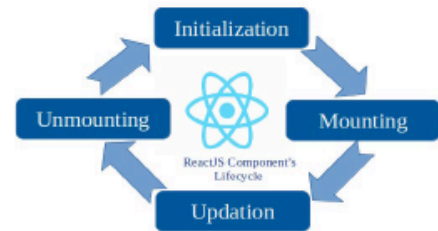
18. Navedeno je nekoliko varijanti kako se zapisuju propsi. Treba izabrati točno napisanu varijantu (ima primjera s vitičastim zagradama, drugačijim redoslijedom riječi itd.)

Props - funkcije

```
function ChildComponent(props) {  
  return <h1>Ja sam {props.name}</h1>;  
}  
  
function ParentComponent() {  
  return (  
    <div>  
      <h1>Ja sam ParentComponent</h1>  
      <ChildComponent name="Child" />  
    </div>  
  );  
}
```

19. Životni ciklus metoda u React-u.

- životni ciklus komponente može se definirati kao niz metoda koje se pozivaju u različitim fazama postojanja komponente
- možemo definirati kod koji će se izvršiti u točno određenom trenutku procesa
- te metode nazivamo *lifecycle methods*, a pozivaju se u sljedećim fazama:
 - inicijalizacija komponente (*initialization*)
 - postavljanje na DOM (*mounting*)
 - ažuriranje (*updating*)
 - brisanje iz DOM-a (*unmounting*)



Neriješeni s ispita:

1. HTML: Izrada korisničkog sučelja aplikacije

Vodeći se temom opisanom u .pdf dokumentu koji se nalazi u mapi koju ste preuzeli, izradite HTML dokument koji sadrži strukturu korisničkog sučelja aplikacije s obrascem i dvije unaprijed pohranjene stavke.

Zadatak riješite u Visual Studio Code-u

2. CSS: Uređivanje korisničkog sučelja aplikacije

Izradite CSS dokument koji uređuje projekt koji ste prethodno izradili na praktičnom zadatku izrade korisničkog sučelja aplikacije.

Neka raspored elemenata korisničkog sučelja odgovara rasporedu elemenata u skici dizajna koju ćete pronaći u .zip datoteci koju ste preuzeli s ispitne aplikacije.

Dizajn za mobilne uređaje mora biti primijenjen na ekranima širine do 768px uključivo, a dizajn za računala mora biti primijenjen na ekranima širine veće od 768px isključivo.

CSS dokument mora biti povezan s projektom kao vanjska lista stilskih pravila.

Izbor boja, tipografija, i ostalih vizualnih elemenata koji se ne tiču rasporeda elemenata na sučelju je proizvoljan.

Media queries breakpoints:

```
/* CSS pravila za zaslone širine manje od 769px */
```

```
@media (max-width: 768px) {  
    CSS pravilo  
}  
  
/* CSS pravilo za zaslone širine 769px i više */  
  
@media (min-width: 769px) {  
    CSS pravilo  
}
```

3. **JavaScript:** Dodavanje funkcionalnosti korisničkom sučelju

Izradite JavaScript dokument koji omogućuje funkcionalnost web aplikacije i korisničkog sučelja, a koje su opisane u kriterijima zadatka. Kriteriji zadatka nalaze se u dokumentu s temom koji ste ranije preuzeli.

JavaScript dokument mora biti povezan s projektom kao vanjska skripta.

(pretpostavljam) Primjer HTML zadatka

Tema: Aplikacija za recenziranje glazbenih albuma

U ovom ispitu kroz zadatke praktičnog tipa izradit ćete web aplikaciju za recenziranje glazbenih albuma.

Aplikacija za recenziranje glazbenih albuma mora ispunjavati sljedeće kriterije:

Kada korisnik otvori web stranicu

Na korisničkom sučelju prikaže se obrazac koji sadrži:

- polje za unos naziva (obavezno polje, tekst)
- polje za unos ocjene (obavezno polje, cijeli broj od 1 do 19),
- dugme za pohranu obrasca

Ako korisnik ispravno popuni obrazac

Kada korisnik pritisne dugme za pohranu obrasca na korisničkom sučelju prikaže se nova stavka koja sadrži:

- naziv
- ocjenu
- datum i vrijeme kada je stavka pohranjena

- dugme za označavanje stavke kao favorita
- dugme za brisanje stavke - ispod te stavke prikazu se na jednak način sve stavke koje je korisnik prethodno pohranio

Ako korisnik neispravno popuni obrazac

Kada korisnik pritisne dugme za pohranu obrasca, preglednik obavještava korisnika da nije ispravno ispunio obrazac.

Kada korisnik odabere dugme za brisanje stavke na korisničkom sučelju, odabrana stavka se više ne prikazuje.

Ako stavka nije označena kao favorit

Kada korisnik odabere dugme za označavanje stavke kao favorita na korisničkom sučelju, stavka se prikazuje tako da je vizualno drugačija od stavki koje nisu istaknute kao favoriti.

Ako je stavka označena kao favorit

Kada korisnik odabere dugme za označavanje stavke kao favorita na korisničkom sučelju, stavka se prikazuje tako da je vizualno drugačija od stavki koje jesu istaknute kao favoriti.