

Zadatak

Na zadanom primjeru objasnite razliku između HTML oznake i atributa:

```
<audio controls src="glazba/jing1.mp3"></audio>
```

HTML oznaka: `audio`

Određuje tip HTML elementa te daje značenje njegovom sadržaju

HTML atributi: `controls`, `src`

Pruža dodatne informacije o sadržaju HTML elementa

Zadatak

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

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

Zadatak

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>
```

Zadatak

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>
```

Zadatak

Zadan je sadržaj:

<https://example.com/movie.mp4>

Pomoću HTML-a objavite video sadržaj s kontrolama za upravljanje reprodukcijom sadržaja. Video formi dodijelite dimenzije 600 x 400 px. (zadatak riješiti u dostupnom code editoru).

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

Zadatak

Zadan je sadržaj na sljedećoj relativnoj putanji:

galerija/panorama-grada.jpg

Pomoću HTML-a objavite grafiku te joj dodijelite dimenzije 720 x 500 px. Postavite zamjenski tekst "Panorama grada" (zadatak riješiti u dostupnom code editoru).

```

```

Zadatak

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

`<footer></footer>`

Zadatak

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

`<article></article>`

Zadatak

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

```
<div></div>
```

Zadatak

Koji element koristimo za tematsko grupiranje sadržaja?

```
<section></section>
```

Zadatak

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

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.)
- više: <https://www.w3.org/WAI/fundamentals/accessibility-principles/>

Zadatak

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

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

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

Zadatak

Nadopunite isječak koda tako da u zaglavlju HTML dokumenta dohvatite vanjsku CSS datoteku sa sljedeće lokacije: `css/style.css`

```
<head>
```

```
<head>
```

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

```
</head>
```

```
</head>
```

Zadatak

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;  
}
```

Zadatak

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

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

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

Zadatak

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

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

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


Zadatak

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

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

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

Zadatak

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

```
{  
  transform: scale(1.1);  
}
```

```
img:hover {  
  transform: scale(1.1);  
}
```

Zadatak

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

```
{  
  color: #eee;  
  font-size: 20px;  
}
```

```
p:first-of-type {  
  color: #eee;  
  font-size: 20px;  
}
```

Zadatak

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;  
}
```

Zadatak

Vrijednost varijable "text-color" postavite kao boju znakova svih paragrafa.

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

```
p {  
  
}
```

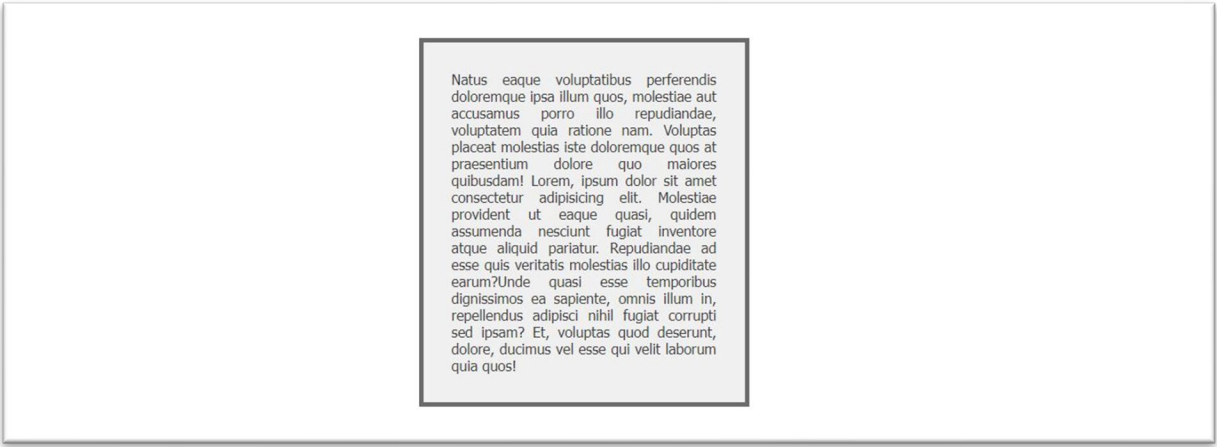
```
:root {  
  --text-color: #333;  
}
```

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

Zadatak

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.



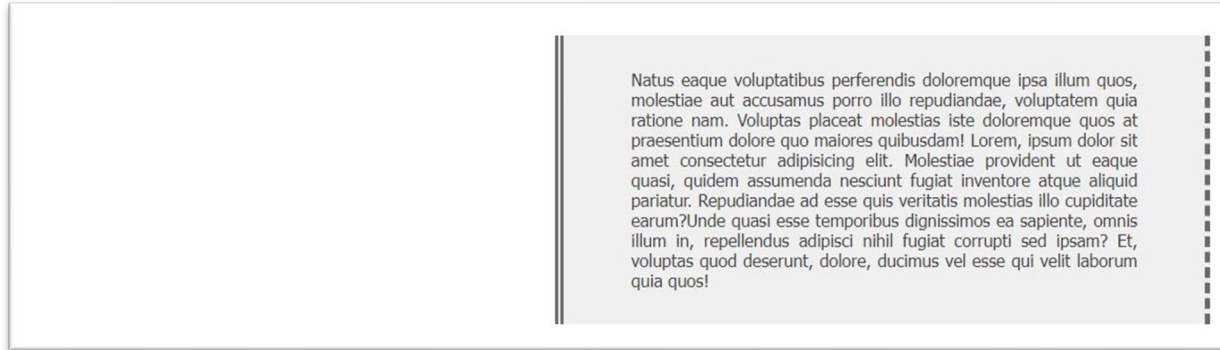
Natus eaque voluptatibus perferendis
doloremque ipsa illum quos, molestiae aut
accusamus porro illo repudiandae,
voluptatem quia ratione nam. Voluptas
placeat molestias iste doloremque quos at
praesentium dolore quo maiores
quibusdam! Lorem, ipsum dolor sit amet
consectetur adipiscing elit. Molestiae
provident ut eaque quasi, quidem
assumenda nesciunt fugiat inventore
atque aliquid pariatur. Repudiandae ad
esse quis veritatis molestias illo cupiditate
earum? Unde quasi esse temporibus
dignissimos ea sapiente, omnis illum in,
repellendus adipisci nihil fugiat corrupti
sed ipsam? Et, voluptas quod deserunt,
dolore, ducimus vel esse qui velit laborum
quia quos!

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

Zadatak

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;  
}
```

Zadatak

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;  
}
```


Zadatak

Unutar pravila koje obuhvaća sve elemente tipa <section> 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;  
}
```

Zadatak

Dohvatite "partials" datoteku sa sljedeće lokacije: base/_color.scss

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

Zadatak

- 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 minimizacija CSS koda.

Zadatak

Je li JavaScript jezik predviđen za objektno-orijentirano ili funkcijsko programiranje?

JavaScript je skriptni programski jezik visoke razine, dobar i za **objektno-orijentirano** i za **funkcijsko programiranje**.

Zadatak

Što je ECMAScript?

JavaScript jezik se razvija prema **ECMAScript specifikaciji**:

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

Zadatak

Što od navedenog nije ispravan identifikator varijable?

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

Zadatak

Koji izraz koristimo u definiciji spremnika za podatke ako želimo onemogućiti promjenu primitivne vrijednosti?

`const`

Zadatak

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 kojemu su definirane
 - moraju se deklarirati prije upotrebe te se ne mogu ponovo deklarirati

Zadatak

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);
```

Zadatak

Deklarirajte varijable y i z te im dodijelite proizvoljne vrijednosti. Nakon toga razlomite ovu ternarnu operaciju na 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;  
}
```

Zadatak

Koji izraz možemo koristiti kao alternativu " if " izraza za postavljanje višestrukog grananja?

Izraz "switch".

Zadatak

Koliko puta će se izvršiti sljedeći blok naredbi?

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

6 puta.

Zadatak

S obzirom da je uvjet petlje lažan blok naredbi se neće niti jedanput izvršiti. Promijenite definiciju uvjetne petlje tako se da blok naredbi uvijek izvrši barem jedanput, bez obzira o istinitosti uvjeta. Vrijednost varijable "count", kao i izraz uvjeta ne smijete promijeniti.

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

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

Zadatak

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.

Zadatak

Napišite JavaScript funkciju koja prihvaća cjelobrojnu vrijednost kao argument te vraća poruku da li se radi o parnom ili neparnom broju. Pretpostavite da će argument uvijek biti cjelobrojna vrijednost.

```
function parniIliNeparni(a){  
    if(a % 2 == 0){  
        return "Broj " + a + " je paran.";  
    }  
    else {  
        return "Broj " + a + " je neparan.";  
    }  
}
```

Zadatak

Napišite JavaScript funkciju koja prihvaća zadani niz kao argument i pronalazi najdužu riječ unutar niza. Vratite tu riječ iz funkcije i spremite u varijablu čiju ćete vrijednost ispisati u konzoli.

```
var rijeci = ['quickest', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog'];
```


Zadatak

```
var rijeći = ['quickest', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog'];  
function pronadiNajduljuRijec(niz) {  
    var rijec = "";  
    for (let i = 0; i < niz.length; i++) {  
        if (rijec.length < niz[i].length) {  
            rijec = niz[i];  
        }  
    }  
    return rijec;  
}  
var najduljaRijec = pronadiNajduljuRijec(rijeći);  
console.log(najduljaRijec);
```

Zadatak

Što predstavlja BOM (Browser Object Model)?

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.

Zadatak

Što predstavlja DOM (Document Object Model)?

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

Definira:

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

Zadatak

Navedite barem tri metode objekta "document" za dohvaćanje HTML elemenata.

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

Zadatak

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);
```

Zadatak

Koristeći JavaScript dohvatite 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);
```

Zadatak

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";
```

Zadatak

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

Odgovor:

Alat koji 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.

Zadatak

Koji tipovi sustava za verzioniranje koda postoje?

Odgovor:

Lokalni sustav za verzioniranje koda

Centralizirani sustav za verzioniranje kôda

Distribuirani sustav za verzioniranje kôda

Zadatak

Što je Git?

Odgovor:

Git je distribuirani sustav za verzioniranje koda (DVCS).

Omogućuje praćenje promjena u izvornom kodu tijekom razvoja softvera te koordiniranje rada među programerima.

Zadatak

Što je repozitorij?

Odgovor:

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.

Zadatak

Napišite Git naredbu s kojom se inicijalizira Git repozitorij.

Rješenje:

```
git init
```

Zadatak

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

Rješenje:

```
git commit -m "message"
```

Zadatak

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

Rješenje:

```
git push
```

Zadatak

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

Rješenje:

```
git pull
```

Zadatak

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

Rješenje:

```
git merge
```


Zadatak

Što predstavlja jQuery?

Odgovor:

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 komunikacija klijenta i poslužitelja (AJAX).

Zadatak

Što je npm (node package manager)?

Odgovor:

Snažan upravitelj programskim paketima:

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

Zadatak

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

Rješenje:

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

ili

```
npm install lodash
```

Zadatak

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).

Rješenje:

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

Zadatak

Što su task runner-i?

Odgovor:

Task runner je alat koji omogućuje automatizaciju ponavljajućih zadataka koji se tipično rade ručno tijekom razvoja projekta.

Primjeri: gulp, grunt, broccoli, brunch, ...

Zadatak

Što je module bundler?

Odgovor:

Alat koji se koristi prilikom izgradnje aplikacije za produkciju.

Omogućuje povezivanje i optimizaciju više modula (datoteka) u jedan proizvodno spreman paket.

Primjeri: webpack, rollup, browserify, ...

Zadatak

Objasnite što je React.

Odgovor:

React je popularna JavaScript biblioteka koja služi za izradu korisničkih sučelja.

Pomoću malih izoliranih dijelova koda (komponenti) stvaramo kompleksno, efikasno i fleksibilno korisničko sučelje.

Zadatak

Napišite naredbu za kreiranje React aplikacije preko CLI-a (*Command-line interface*). Naziv aplikacije glasi "react-app".

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


Zadatak

Što je potrebno napraviti kako bi se programski kôd 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

Zadatak

Prikazana je definicija React komponente koja sadrži grešku u 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

Zadatak

Što je potrebno upisati na prazno mjesto kako bi uključili JavaScript izraz unutar JSX 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

Zadatak

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>;  
}
```

Zadatak

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>;  
}
```

Zadatak

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

Odgovor:

Komponente je moguće kreirati pomoću funkcije ili klase.

Zadatak

Objasnite što su propsi ("property-i").

Propsi su način na koji komponente međusobno komuniciraju:

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

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

- komponenta dijete ne može i ne smije mijenjati props objekt koji je primila od roditelja.

Zadatak

Što je potrebno upisati na prazno mjesto kako bi komponenti "ChildComponent" proslijedili props "name" s proizvoljnom vrijednošću (npr. "Child")?

```
function ParentComponent() {  
  return (  
    <div>  
      <h1>Ja sam ParentComponent</h1>  
      <ChildComponent _____ />  
    </div>  
  );  
}
```

```
function ChildComponent(props) {  
  return <h1>Ja sam {props.name}</h1>;  
}
```


Odgovor

Što je potrebno upisati na prazno mjesto kako bi komponenti "ChildComponent" proslijedio props "name" s proizvoljnom vrijednošću (npr. "Child")?

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

```
function ChildComponent(props) {  
  return <h1>Ja sam {props.name}</h1>;  
}
```

Zadatak

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 prikazala promjene u stanju.

Ponovnim renderiranjem komponente, renderiraju se i sve njezine child komponente.

Zadatak

Putem koje metode je jedino ispravno mijenjati stanje komponente definirane klasom, kako bi osigurali da se ona ponovno iscrta (renderira)?

Odgovor:

Metoda `setState`.

Zadatak

Što omogućuju hook funkcije? Navedite nekoliko primjera hook funkcija.

Hook funkcije omogućuju upotrebu stanja (state) i drugih mogućnosti unutar komponenata 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.

Zadatak

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

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

Odgovor

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

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

Zadatak

Navedite faze životnog ciklusa React komponente.

1. Inicijalizacija komponente (initialization)
2. Postavljanje na DOM (mounting)
3. Ažuriranje (updating)
4. Brisanje iz DOM-a (unmounting)

Zadatak

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 bolje performanse.

Zadatak

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

1. CSS Stylesheet
2. Inline stilovi
3. CSS Modules

Zadatak

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.