

1. লোকাল ডেভেলপমেন্ট এনভায়রনমেন্ট সেটআপে কোন প্রোটোকল ব্রাউজার ব্যবহার করে?

✓ উত্তর: HTTP

ব্যাখ্যা: ব্রাউজার সাধারণত HTTP ব্যবহার করে ওয়েব পেজ লোড করতে।

2. কোন HTML এলিমেন্ট ন্যাভিগেশন লিঙ্কের জন্য ব্যবহৃত হয়?

✓ উত্তর: <nav>

ব্যাখ্যা: <nav> ট্যাগ ওয়েবসাইটের মেনু/লিঙ্ক সেকশনের জন্য ব্যবহৃত হয়।

3. এবং ট্যাগের পার্থক্য কী?

✓ উত্তর: purely visual, semantic importance

ব্যাখ্যা: গুরুত্বপূর্ণ টেক্সট বোঝায়, শুধু বোল্ড দেখায়।

4. Flexbox-এ কোন প্রপার্টি টেক্সট আইটেমকে প্রয়োজন হলে সংকুচিত হতে দেয়?

✓ উত্তর: flex-shrink

ব্যাখ্যা: এটি flex item কে সংকুচিত হওয়ার অনুমতি দেয়।

5. CSS Grid-এ repeat(auto-fit, minmax(200px, 1fr)) কী করে?

✓ উত্তর: Creates flexible columns that fill available space

ব্যাখ্যা: যতটা জায়গা ফাঁকা থাকে তা অনুযায়ী গ্রিড কলাম স্বয়ংক্রিয়ভাবে সামঞ্জস্য হয়।

6. Bootstrap বা Tailwind-এর মতো CSS ফ্রেমওয়ার্কের সুবিধা কী?

✓ উত্তর (সংক্ষিপ্ত): দ্রুত UI ডিজাইন, রেস্পন্সিভ গ্রিড, রেডিমেড ক্লাস।

ব্যাখ্যা: কোড কম লিখে দ্রুত রেস্পন্সিভ ওয়েবসাইট বানানো যায়।

প্রশ্ন 7:

রাস্টার (Raster) এবং ভেক্টর (Vector) ইমেজের মধ্যে পার্থক্য কী?

উত্তর:

রাস্টার ইমেজ ছোট ছোট পিক্সেল দিয়ে তৈরি এবং রেজোলিউশন-নির্ভর। বড় করলে মান নষ্ট হয়। উদাহরণ: JPG, PNG।

ভেক্টর ইমেজ গাণিতিক সূত্রের মাধ্যমে তৈরি হয়, তাই বড় করলে মান নষ্ট হয় না। উদাহরণ: SVG, EPS।

প্রশ্ন ৪:

CSS-এ Absolute Units এবং Relative Units এর মধ্যে পার্থক্য কী?

উত্তর:

- **Absolute Units:** নির্দিষ্ট মাপ, যা পরিবর্তন হয় না। যেমন: px, pt, cm, mm।
 - **Relative Units:** পরিবেশ বা প্যারেন্ট উপাদানের উপর নির্ভর করে মাপ পরিবর্তন হয়। যেমন: %, em, rem, vw, vh।
-

প্রশ্ন ৯:

JavaScript-এ forEach() মেথড কী? এবং এটি কীভাবে একটি অ্যারে লুপ করতে ব্যবহৃত হয়?

উত্তর:

forEach() হলো JavaScript-এর একটি অ্যারে মেথড, যা অ্যারের প্রতিটি উপাদানের ওপর একটি করে ফাংশন চালাতে ব্যবহৃত হয়।

```
arr.forEach((item) => {  
    console.log(item);  
});
```

প্রশ্ন 10:

JavaScript-এ addEventListener() কী এবং এটি onclick অ্যাট্রিবিউট থেকে কীভাবে ভিন্ন?

উত্তর:

addEventListener() একটি এলিমেন্টের ওপর ইভেন্ট হ্যান্ডলার যুক্ত করতে ব্যবহৃত হয়। এটি

প্রশ্ন 11:

JavaScript-এ `forEach()` এবং `map()` মেথডের মধ্যে পার্থক্য কী?

উত্তর:

- `forEach()`: কেবল প্রতিটি উপাদানের ওপর অ্যাকশন চালায়, নতুন অ্যারে রিটার্ন করে না।
- `map()`: প্রতিটি উপাদান প্রসেস করে একটি নতুন অ্যারে রিটার্ন করে।

প্রশ্ন 12:

একটি মেইল মার্জ (Mail Merge) কীভাবে কাজ করে এবং এর সাধারণ ব্যবহার কী?

উত্তর:

মেইল মার্জ মূল ডকুমেন্ট (যেমন: চিঠি) এবং একটি ডেটা সোর্স (যেমন: নামের তালিকা) একত্র করে স্বয়ংক্রিয়ভাবে ব্যক্তিভেদে আলাদা আলাদা ডকুমেন্ট তৈরি করতে ব্যবহৃত হয়। সাধারণত এটি ব্যক্তিগতকৃত চিঠি, আমন্ত্রণপত্র, ফলাফল শিট ইত্যাদি তৈরিতে ব্যবহৃত হয়।

প্রশ্ন 13:

একটি ওয়েব ডিজাইন প্রজেক্টের মৌলিক ফোল্ডার স্ট্রাকচার তৈরি করতে কোন ধাপগুলো থাকে?

উত্তর:

একটি সাধারণ ওয়েব ডিজাইন প্রজেক্টে সাধারণত নিচের মতো ফোল্ডার স্ট্রাকচার থাকে—

- `project-name/`
 - `index.html`
 - `css/` (স্টাইল ফাইল)
 - `scripts/` (জাভাস্ক্রিপ্ট ফাইল)

এগুলো ওয়েবসাইটকে সংগঠিত এবং ব্যবস্থাপনা সহজ করে।

প্রশ্ন 14:

সেকশন (`<section>`) এবং ডিভ (`<div>`) এলিমেন্টের মধ্যে পার্থক্য কী?

উত্তর:

- **section:** এটি অর্থপূর্ণ কনটেন্ট বা টপিক অনুযায়ী অংশ ভাগ করতে ব্যবহৃত হয় এবং এটি সেমান্টিক এলিমেন্ট।
 - **div:** এটি সাধারণ কন্টেন্টের হিসেবে ব্যবহৃত হয় এবং সেমান্টিক নয়। মূলত স্টাইলিং বা লেআউট গ্রুপিং-এর জন্য ব্যবহৃত হয়।
-

প্রশ্ন 15:

ওয়ার্কপ্লেস এনভায়রনমেন্টে ICT ব্যবহারের উদ্দেশ্য কী?

উত্তর:

ICT ব্যবহারের উদ্দেশ্য হলো তথ্য প্রযুক্তির দায়িত্বশীল ও কার্যকর ব্যবহার নিশ্চিত করা—যেমন নিরাপত্তা বজায় রাখা, প্রাইভেসি রক্ষা করা এবং সঠিকভাবে যোগাযোগ পরিচালনা করা।

প্রশ্ন 16:

নিম্নলিখিত জাভাস্ক্রিপ্ট কোডের আউটপুট কী হবে?

```
var a = 5, b = 10;  
console.log(b == 10 && a == 5);  
console.log(a > b || b == 10);  
console.log(!(a == 5));
```

উত্তর:

- true
- true
- false

প্রশ্ন 17:

রাস্টার (Raster) এবং ভেক্টর (Vector) ইমেজের মধ্যে পার্থক্য ব্যাখ্যা করুন, এবং উদাহরণ দিন।

উত্তর:

- **রাস্টার ইমেজ:** অসংখ্য ছোট পিক্সেল দিয়ে তৈরি হয়। বড় করলে মান নষ্ট হয়।
উদাহরণ: JPEG, PNG।

- **ভেক্টর ইমেজ:** গাণিতিক সূত্র ও আকৃতি ব্যবহার করে তৈরি হয়। বড় করলেও মান নষ্ট হয় না।

উদাহরণ: SVG, AI, EPS।

প্রশ্ন 18:

CSS-এ Relative units (যেমন: em, rem) এবং Absolute units (যেমন: px, pt, cm, mm)-এর মধ্যে পার্থক্য কী?

উত্তর:

- **Absolute units:** নির্দিষ্ট এবং স্থির; স্কেলে পরিবর্তন হয় না।
যেমন: px, pt, cm, mm
 - **Relative units:** অন্য কিছুর উপর নির্ভর করে; যেমন প্যারেন্ট বা root element-এর সাইজ।
যেমন: em, rem, %, vw, vh
-

প্রশ্ন 19:

Array কী? এবং JavaScript-এ forEach() ব্যবহার করে কীভাবে একটি অ্যারের মধ্য দিয়ে লুপ করা যায়?

উত্তর:

Array হলো একাধিক মানকে একসাথে সংরক্ষণ করার একটি ডেটা স্ট্রাকচার।
forEach() মেথড অ্যারের প্রতিটি উপাদানের জন্য একটি ফাংশন চালায়।

উদাহরণ:

```
let arr = [1, 2, 3];  
arr.forEach(item => console.log(item));
```

প্রশ্ন 20:

JavaScript-এ addEventListener() মেথডের ব্যবহার কী? এবং এটি onclick থেকে কীভাবে ভিন্ন?

উত্তর:

- `addEventListener()` কোনো নির্দিষ্ট এলিমেন্টে ইভেন্ট হ্যান্ডলার যোগ করতে ব্যবহৃত হয়। এটি একাধিক ইভেন্ট যোগ করার সুবিধা দেয়।
 - `onclick` শুধুমাত্র একটি ইভেন্ট হ্যান্ডলার সেট করতে পারে এবং একই এলিমেন্টে একাধিক ইভেন্ট সহজে যোগ করা যায় না।
-

প্রশ্ন 21:

JavaScript-এর `forEach()` এবং `map()` মেথডের উদ্দেশ্য ব্যাখ্যা করুন। এদের আউটপুটের পার্থক্য কী?

উত্তর:

- `forEach()`: অ্যারের প্রতিটি উপাদানের ওপর কাজ করে কিন্তু নতুন কোনো অ্যারে রিটার্ন করে না।
- `map()`: প্রতিটি উপাদান প্রসেস করে একটি নতুন অ্যারে রিটার্ন করে।

প্রশ্ন 22:

JavaScript-এ `==` এবং `===` অপারেটরের মধ্যে পার্থক্য কী? এবং কোনটি কেন বেশি ব্যবহার করা উচিত?

উত্তর:

- `== (Loose Equality)`: ডেটা টাইপ আলাদা হলেও মান মিললে `true` রিটার্ন করে। অর্থাৎ এটি টাইপ কনভার্সন করে।
- `=== (Strict Equality)`: মান এবং ডেটা টাইপ – উভয়ই একই হলে `true` রিটার্ন করে। কোনো টাইপ কনভার্সন করে না।

কেন `===` বেশি ব্যবহার করা উচিত?

- এটি নিরাপদ এবং পূর্বানুমানযোগ্য ফল দেয়।
 - টাইপ সংক্রান্ত ভুল কমায়।
-

প্রশ্ন 23:

Flexbox কীভাবে রেসপনসিভ ওয়েব লেআউট তৈরি করতে সহায়তা করে?

উত্তর:

Flexbox ওয়েব এলিমেন্টগুলিকে স্বয়ংক্রিয়ভাবে স্থান বণ্টন, আকার পরিবর্তন ও সঠিকভাবে সাজাতে সাহায্য করে।

রেসপনসিভ ডিজাইনে Flexbox-এর সুবিধাসমূহ:

- কন্টেইনারের ভেতরে উপাদানগুলো সহজে সেন্টার বা অ্যালাইন করা যায়।
 - স্ক্রিন সাইজ অনুযায়ী আইটেম স্বয়ংক্রিয়ভাবে লাইনে বা কলামে সাজে।
 - আইটেমের ফাঁকা জায়গা (space-between, space-around) সহজে নিয়ন্ত্রণ করা যায়।
 - মোবাইল, ট্যাবলেট ও ডেস্কটপ লেআউট সহজে মানিয়ে যায়।
-

প্রশ্ন 24:

CSS Framework ব্যবহার করার প্রধান সুবিধা ও অসুবিধা কী? কাস্টম CSS কেন কিছু ক্ষেত্রে ভালো?

উত্তর:

সুবিধা:

- দ্রুত ও সহজে লেআউট তৈরি করা যায়।
- রেসপনসিভ ডিজাইন প্রস্তুত থাকে।
- সাধারণ কম্পোনেন্ট (button, form, grid) আগে থেকেই পাওয়া যায়।
- কোডের ধারাবাহিকতা বজায় থাকে।

অসুবিধা:

- অনাবশ্যক কোড লোড হয়ে ওয়েবসাইট ভারী হতে পারে।
- কাস্টমাইজেশন কখনও কঠিন হতে পারে।
- সব ডিজাইন একই রকম দেখানোর ঝুঁকি থাকে।

কেন কাস্টম CSS ভাল হতে পারে?

- সম্পূর্ণ ডিজাইন নিয়ন্ত্রণ নিজের হাতে থাকে।
- অতিরিক্ত কোড লোড হয় না, ফলে পেজ লাইটওয়েট হয়।

প্রশ্ন 25:

CSS-এ position: absolute, position: relative এবং position: fixed—এদের মধ্যে পার্থক্য ব্যাখ্যা করুন।

উত্তর:

1. position: relative

- এলিমেন্ট তার স্বাভাবিক অবস্থান থেকে relativeভাবে সরানো যায়।
- এর অবস্থান পরিবর্তন হলেও এলিমেন্টের আসল জায়গাটি দখল করা থাকে।

2. position: absolute

- নিকটস্থ positioned প্যারেন্টের relative হয়ে অবস্থান নির্ধারণ করে।
- স্বাভাবিক লেআউট থেকে বিচ্ছিন্ন হয়ে যায়।

3. position: fixed

- ভিউপোর্টের সাথে স্থির থাকে।
- স্ক্রল করলেও অবস্থান পরিবর্তন হয় না।
- সাধারণত ন্যাভিগেশন বার বা ফ্ল্যাটিং বাটনের জন্য ব্যবহৃত হয়।

প্রশ্ন 26:

ওয়ার্কপ্লেস পরিবেশে ICT ব্যবহারের উদ্দেশ্য কী?

উত্তর:

ওয়ার্কপ্লেসে ICT ব্যবহারের প্রধান উদ্দেশ্য হলো—

- তথ্য ও প্রযুক্তির দায়িত্বশীল ব্যবহার নিশ্চিত করা,
 - নিরাপত্তা (Security) বজায় রাখা,
 - ডেটা প্রাইভেসি রক্ষা করা,
 - কার্যকর যোগাযোগ পরিচালনা করা,
 - এবং ডিজিটাল রিসোর্সের সঠিক ব্যবস্থাপনা নিশ্চিত করা।
-

প্রশ্ন 27:

Mail Merge কীভাবে কাজ করে এবং এর ব্যবহার কোথায়?

উত্তর:

Mail Merge হলো একটি ওয়ার্ড-প্রসেসিং সুবিধা যেখানে—

- একটি মূল ডকুমেন্ট (যেমন: চিঠি) এবং
- একটি ডেটা সোর্স (যেমন: নামের তালিকা, ঠিকানা) একত্র করে স্বয়ংক্রিয়ভাবে ব্যক্তিগতকৃত ডকুমেন্ট তৈরি করা হয়।

সাধারণ ব্যবহার:

- আমন্ত্রণপত্র
- রেজাল্ট শিট
- কাস্টমাইজড চিঠি
- সনদ বা স্লিপ তৈরি

প্রশ্ন 28:

একটি ওয়েব ডিজাইন প্রজেক্টের মৌলিক ফোল্ডার স্ট্রাকচার তৈরি করতে কোন ধাপগুলো অনুসরণ করা হয়?

উত্তর:

ওয়েব ডিজাইন প্রজেক্টের জন্য সাধারণত নিচের মতো ফোল্ডার স্ট্রাকচার তৈরি করা হয়—

project-name/

|

|— index.html

|

|— css/

| └— style.css

|

└─ scripts/

└─ script.js

এই স্ট্রাকচার কোড পরিচালনা, ফাইল আলাদা করা এবং প্রজেক্টকে পরিষ্কারভাবে সংগঠিত রাখতে সাহায্য করে।

প্রশ্ন 29:

section, article এবং div—এই HTML এলিমেন্টগুলোর মধ্যে পার্থক্য কী?

উত্তর:

- **section:**
 - বড় কোনো টপিক বা কনটেন্টের অংশ ভাগ করতে ব্যবহৃত হয়।
 - এটি সেমান্টিক (অর্থপূর্ণ)।
- **article:**
 - স্বতন্ত্রভাবে পড়া যায় এমন কনটেন্ট (যেমন: ব্লগ পোস্ট, নিউজ আর্টিকেল) এর জন্য ব্যবহৃত হয়।
 - এটি সেমান্টিক এলিমেন্ট।
- **div:**
 - কোনো সেমান্টিক অর্থ নেই, শুধু লেআউট বা গ্রুপিং-এর জন্য ব্যবহৃত হয়।
 - সাধারনত CSS বা স্ক্রিপ্টিং-এর সঙ্গে ব্যবহৃত হয়।

30. What will be logged?

```
Const { a=5, b=10}={a:7};
```

```
Console.log(a,b);
```

- a) 5 10 **b) 7 10** c) 7 undefined d) undefined 10

31. What will be the code print?

```
async function fetchData() {
```

```
  console.log('1');
```

```
  await Promise.resolve();
```

```
console.log('2');  
}
```

```
fetchData();  
console.log('3');
```

ব্যাখ্যা — Event Loop অনুযায়ী

1. fetchData() চালানোর সাথে সাথে synchronous কোড চলবে → '1'
2. await Promise.resolve() → মাইক্রোটাস্ক কিউতে যায়
(এসময় জাভাস্ক্রিপ্ট পরের synchronous কোড চালায়)
3. পরের synchronous লাইন → '3'
4. মাইক্রোটাস্ক কিউ সম্পন্ন হয় → '2'

a) 1 2 3

b) 1->3->2

c) 3 1 2

d) 2 1 3