

# Interview Questions & Answers – Last 1 Year Preparation Guide

Focus: **Java + Playwright + Cypress + API + Git + Testing Concepts**

Format: **Question → Short, Interview-ready Answer / Code**

---

## 1 Java & Programming (Most Asked)

### Q1. Filter prices using Java Streams

**Problem:** Given list of WebElements with prices like "\$1,200.50", filter > 500 and sort.

```
List<Double> prices = elements.stream()
    .map(e -> e.getText().replace("$", "").replace(",", ""))
    .map(Double::parseDouble)
    .filter(p -> p > 500)
    .sorted()
    .collect(Collectors.toList());
```

---

### Q2. Remove spaces from string

```
String s = "I Am Selenium Test Engineer";
String out = s.replaceAll("\\s", "");
```

---

### Q3. Character occurrence count

```
String name = "Aautttommation";
Map<Character,Integer> map = new HashMap<>();
for(char c : name.toCharArray()){
    map.put(c, map.getOrDefault(c, 0) + 1);
}
System.out.println(map);
```

---

### Q4. Missing numbers between min & max

```
int[] arr = {5,7,10};
int min = Arrays.stream(arr).min().getAsInt();
int max = Arrays.stream(arr).max().getAsInt();
```

---

```
Set<Integer> set = Arrays.stream(arr).boxed().collect(Collectors.toSet());
for(int i=min;i<=max;i++){
    if(!set.contains(i)) System.out.print(i + " ");
}
```

---

#### Q5. String contains only digits

```
boolean isNumber = str.matches("\\d+");
```

---

## 2 Playwright – Core Interview Q&A

#### Q6. Can we use JavaScriptExecutor in Playwright?

✗ No. Playwright directly supports JS execution:

```
await page.evaluate(() => document.title);
```

---

#### Q7. Run tests in Firefox

```
npx playwright test --project=firefox
```

---

#### Q8. Parallel execution (7 workers)

```
npx playwright test --workers=7
```

---

#### Q9. test.describe usage

Used for **grouping tests** and controlling execution mode.

```
test.describe('group', () => { ... })
```

---

#### Q10. Wait for download

```
const download = await Promise.all([
    page.waitForEvent('download'),
```

```
page.click('#download')
]);
await download[0].saveAs('file.pdf');
```

---

### Q11. Sequential + Parallel tests

```
test.describe.serial('serial tests', () => { ... })
```

Other test files run in parallel.

---

### Q12. Handle SSL error

```
chromium.launch({ ignoreHTTPSErrors: true });
```

---

### Q13. Lazy loading (LinkedIn type)

```
await page.mouse.wheel(0, 3000);
```

---

## 3 Cypress Interview Questions

### Q14. Sync vs Async

- Cypress: **Command queue (sync-looking)**
- Playwright: **True async/await**

---

### Q15. Handle alert / mouse / window

```
cy.on('window:alert', msg => {})  
cy.rightclick()
```

---

### Q16. Advantage over Selenium

- Auto wait
- Time travel
- Fast debugging

## 4 API Automation

### Q17. PUT request – record not found

→ 404 Not Found

---

### Q18. Print API response to file

```
Files.write(Paths.get("resp.json"), response.asString().getBytes());
```

---

### Q19. API + UI in same project?

✓ Yes – best practice for **setup & cleanup**

---

## 5 Git – Frequently Asked

### Q20. Rebase vs Merge

- Rebase: clean history
- Merge: keeps commits

### Q21. Solve rebase conflict

```
git rebase master  
git add .  
git rebase --continue
```

### Q22. Get remote master

```
git fetch origin  
git checkout master  
git pull origin master
```

---

## 6 Testing Concepts

### Q23. Shift Left vs Shift Right

- Shift Left: testing early (unit, API)
  - Shift Right: monitoring in prod
-

## Q24. Definition of Done

Feature is **coded + tested + reviewed + deployed + documented**

---

## Q25. Accessibility testing

- Axe
  - Lighthouse
  - WCAG rules
- 

## Q26. Visual testing

- Playwright screenshot
  - Percy / AppliTools
- 

## 7 Environment & Config

### Q27. .env usage

Best for **secrets**, not role logic. Roles → config files or fixtures.

---

### Q28. BDD vs Normal Playwright

- BDD: readable, slower
  - Normal: faster, flexible
- 

## 8 CI / Debugging

### Q29. Local pass, CI fail

- Headless issues
  - Timing
  - Env mismatch
- 

### Q30. Mock API

Used to **remove dependency & stabilize tests**

---

## 9 Mindset Questions (Very Important)

### Q31. Why Cypress / Playwright came?

To solve **flakiness, speed, async issues** of Selenium.

### Q32. Learning approach

- Understand internals
  - Practice daily
  - Ask **WHY** always
- 

## 10 Final Interview Strategy

- ✓ Logic first
  - ✓ WHY explanation
  - ✓ Real project examples
  - ✓ API + UI optimization stories
- 

✓ This document covers **90% questions asked in last 1 year interviews**