**MAY EXAMS**

**1. What utility class adds horizontal padding in Tailwind?**  
The correct answer is px-4. This class applies horizontal (left and right) padding of 1rem (or 4 units). I remember struggling to memorize these tiny utility classes, but now I understand the pattern: p is for padding, x for horizontal, and the number controls the size.

**2. Which property defines the ability for a flex item to grow if necessary?**  
The right answer is flex-grow. This property controls whether a flex item can grow to fill extra space in its flex container. I used to mix it up with flex-basis or flex-shrink, but now I know flex-grow specifically handles extra space.

**3. Which command initializes a new Git repository?**  
The answer is git init. Running this command inside a folder starts Git tracking. I remember the first time I typed this — it felt like stepping into a professional dev world!

**4. What command stages all changes for commit?**  
The answer is git add .. This command adds all the modified and new files in the directory to the staging area. It’s easy to forget the dot (.), but it’s essential.

**5. Which media query is correct for screens smaller than 768px?**  
The correct media query is @media (max-width: 768px). I had to learn that max-width targets devices up to that width, which is crucial for mobile-first design.

**6. What is GitHub?**  
GitHub is a cloud-based Git repository hosting service. It’s where I store code, collaborate, and track changes online — a lifesaver when working across devices or with teams.

**7. Which unit is relative to the width of the viewport?**  
The answer is vw (viewport width). I used to confuse vw with vh (viewport height), but now I know vw is perfect for fluid, responsive layouts.

**8. What does grid-column: 1 / 3; do?**  
It spans the element from column line 1 to 3. In simpler terms, the item takes up two columns in a grid. It took me a while to visualize grid lines, but drawing them out really helped.

**9. What does git commit -m "message" do?**  
This command creates a new commit with a message describing the changes. I now appreciate how important good commit messages are for future me (and my team).

**10. What property controls the alignment of flex items along the main axis?**  
The answer is justify-content. It lets me align flex items horizontally (if flex-direction: row) or vertically (if flex-direction: column). Getting the axis right was a bit confusing at first!

**11. What is the difference between relative and absolute positioning?**  
Relative positions the element relative to its normal spot, while absolute removes the element from the document flow and positions it relative to the nearest positioned ancestor. I remember breaking layouts before I truly understood this!

**12. What does git push do?**  
It uploads local repository commits to a remote repository, like GitHub. The first time I pushed code felt so nerve-wracking — like exposing my work to the world.

**13. What value of justify-content spaces items evenly with space around them?**  
The correct value is space-around. This puts equal space on both sides of each item. Understanding this made my layouts feel polished and balanced.

**14. What is the default value of the flex-direction property in Flexbox?**  
The default is row. This means flex items are laid out horizontally unless I explicitly change it. It’s one of those things you forget until something breaks!

**15. What does git clone do?**  
It creates a copy of a remote repository locally. This command opened the door for me to explore other people’s projects and learn from real codebases.

**16. What does position: sticky do?**  
It toggles between relative and fixed positioning, depending on scroll. I love using it for sticky headers — it’s like magic when done right!

**17. Which property controls the space between grid items?**  
The gap property. It replaced grid-gap and applies space between both rows and columns. This simplified so much compared to using margins.

**18. What command lists the commit history?**  
The command is git log. Scrolling through this history helped me understand the evolution of a project — and sometimes laugh at my own early mistakes.

**19. What does flex: 1 0 auto; mean?**  
It’s a shorthand for flex-grow: 1, flex-shrink: 0, and flex-basis: auto. It lets an item grow to fill space without shrinking below its content size. This one took me a few rereads to grasp fully.

**20. How do you make a flex container wrap its items into multiple lines?**  
You use flex-wrap: wrap;. I remember the joy when my overflowing flex items neatly wrapped onto a new line for the first time!