Starting your own website involves a combination of **planning**, **designing**, and **coding**. Here’s a structured approach to help you get started:

**1. Define Your Purpose and Goals**

* **Purpose**: What is the website for? (e.g., personal blog, portfolio, business site)

**Purpose: Portfolio**

* Goals: What do you want to achieve with it? (e.g., showcase work, sell products, share information)

**Goals: Showcase work**

**2. Research and Inspiration**

* **Look at Examples:** Browse other websites for inspiration. Note what you like and dislike.

https://www.aliciawileyphotography.com/

* **Identify Your Audience: Who** will be visiting your website? Tailor your design and content accordingly.

Personas que necesitan fotos para eventos como bodas, fiestas de cumpleaños, quinceaños, etc.

**3.** **Planning and Design**

* **Create a Sitemap:** Outline the structure of your website. Decide what pages you need (e.g., Home, About, Contact).

**HOME, ABOUT, EXPERIENCE, PORTFOLIO, ALBUMS, CONTACTO, PHOTO BOOTH**

* **Wireframes:** Sketch a basic layout of your pages. Focus on the placement of elements without worrying about the details.
* **Design Mockups:** Use a design tool (e.g., Adobe XD, Figma, Sketch) to create detailed mockups of your pages.

**4. Choose Your Tools and Technology**

* **HTML/CSS**: Learn the basics of HTML and CSS for structuring and styling your website.
* **JavaScript**: For interactivity and dynamic content.
* **Frameworks**: Consider using frameworks/libraries like Bootstrap (for responsive design) or React (for dynamic web apps).
* **Content Management Systems (CMS)**: For larger projects, consider using WordPress, Joomla, etc.

**5. Set Up Your Development Environment**

* **Text Editor/IDE**: Choose a code editor like VS Code, Sublime Text, or Atom.
* **Local Server**: Use tools like XAMPP or MAMP to run your website locally.
* **Version Control**: Use Git for version control and GitHub for collaboration and backup.

**6. Start Coding**

* **HTML Structure**: Begin by coding the basic structure of your website using HTML.
* **Styling with CSS**: Add styles to your HTML using CSS. Consider using CSS preprocessors like Sass for better organization.
* **Add Interactivity**: Use JavaScript to add interactivity to your site. Start with simple scripts and build up.

**7. Responsive Design**

* **Media Queries**: Ensure your website is responsive and works on different devices by using CSS media queries.
* **Testing**: Test your site on various devices and browsers to ensure compatibility.

**8. Content Creation**

* **Text and Images**: Add relevant content, including text, images, and multimedia.
* **SEO**: Optimize your content for search engines by using appropriate tags, keywords, and meta descriptions.

**9. Testing and Debugging**

* **Usability Testing**: Get feedback from users to identify any usability issues.
* **Debugging Tools**: Use browser developer tools to debug and fix any issues.

**10. Deployment**

* **Domain and Hosting**: Purchase a domain name and choose a hosting provider.
* **Upload Your Website**: Use FTP or deployment tools to upload your website to the server.

**11. Maintenance**

* **Updates**: Regularly update your content and software.
* **Security**: Ensure your website is secure by keeping software updated and using security best practices.

**Summary**

1. **Plan**: Define purpose, goals, and audience.
2. **Design**: Create sitemaps, wireframes, and mockups.
3. **Choose Tools**: Select tools and technologies for development.
4. **Set Up**: Prepare your development environment.
5. **Code**: Start with HTML, then CSS, and JavaScript.
6. **Responsive**: Ensure the site works on all devices.
7. **Content**: Add and optimize your content.
8. **Test**: Conduct thorough testing.
9. **Deploy**: Publish your website.S
10. **Maintain**: Regular updates and maintenance.

Starting with a clear design and plan will make the development process smoother and more efficient.

**FLORWERSHOTSV**

Purpose: Portfolio