I'm not robot

reCAPTCHA

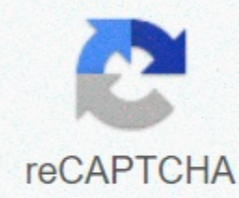**Continue**

Web software development

An Integrated Development Environment (IDE) is software that consolidates the basic tools needed for software testing and writing. Without IDE, a developer would have to select and manage all these tools separately, but IDE brings all these tools together as a single framework or service. Needless to say, IDE helps a web developer's work to be more precise and quicker. In this guide, we will introduce you to 10 best IDE web development options, which include feature-rich support with multiple code languages. But before starting, let's answer the common question"What is an IDE?" in more detail as well as how to pick the best IDE for your specific needs. Let's dig in. What is an IDE? An Integrated Development Environment (IDE) is a software that you can load onto your computer. They're designed to streamline the web development process as mentioned earlier. An IDE includes these main features: Text editor – An IDE is a text editor that includes several other specific features. Compiler or Interpreter – An integrated program that translates your code into a computer-readable language. This lets your code run, referred to as having your script executed. An interpreter is a program that's included. Its purpose is to execute code without it needing to be compiled first. Build or Make Integration – These are included ways to automate necessary processes. Debugger – An included program that tests your code for errors so you can resolve them. Syntax Highlighter – Parts of your code are displayed in colors that correspond with different elements such as properties, tags, attributes, and all the rest. This helps make your code visually easier to scan and see where you made small errors such as forgetting to close a tag. Graphical User Interface (GUI) – This is a screen that's human-readable and that's straightforward to navigate as well as accomplish tasks. It can include buttons, a menu, and other similar elements instead of just text and the option to enter a text command. Other possible features – Many IDEs develop and include new features that other IDEs may not offer. How to Find the Right IDE for You When deciding on an IDE that you want to use, it's essential that you first determine what you need out of it. This is so because each IDE can differ between the features that are offered. That also means your favorite IDE may differ from that of your peers so just pick the one that works for you at the end of the day. Features You May Need in Your Pick for Best IDE Here's a short list of the possible features you may need outside of the typical components of an IDE as listed earlier: Programming language support – The IDE you choose should support the code languages you need for your projects. Customizable text editors – Some IDEs offer the ability to edit the GUI. Unit testing – This is the option of being able to add mock objects to sections of your code so it can be tested straight away without having to complete that section first. Source code library – Some IDEs have resources that include scripts and source code you can use. Error diagnostics and reports – If you need errors to be detected and listed for you, then this is a feature to consider in your pick for best IDE web development projects need. Code completion – Many IDEs can intelligently finish your code for you. For example, an IDE can detect when you're ready to close a tag and will close it for you to save you time. Integrations and plugins – with GitHub and Apache's Subversion Code search – You can search through your code quickly for whatever element you want. Hierarchy diagrams – Your files can be set up and displayed in a hierarchy for larger projects containing many files and scripts that all work together in a specific way. Displaying a hierarchy diagram can help you see the order in which files are executed as well as which files and scripts are related to each other. Model-driven development (MDD) – This is a process where a developer creates a model of what they want to code, then the IDE helps you code it. After that, the IDE automates debugging and testing to some degree so it's easier for you to do some more debugging and testing on your own. Be sure to make note of what features you need when you check out the best IDE for web development below. The 10 Best IDE for Web Development This list focuses on the most popular, powerful, feature-rich, and as a result, the best IDE options. 1. Visual Studio The Visual Studio IDE is one of the most popular and best IDE web development options available. It uses AI to learn from your edits as you code so it can finish your sentences – er, lines of code. On top of that, you can collaborate with your team, live, when you're editing and debugging. You can also share servers, terminals, and comments. Visual Studio supports web, mobile, app, and game development, ASP.NET, Python, Node.js, C++, Unity as well as support for Azure. You can also create development environments in the cloud and a lot more while being available for Windows, Mac, Android, iOS, web, and in the cloud. 2. IntelliJ IDEA IntelliJ IDEA is certainly one of the best IDE for Java. It focuses on features to help you be as productive as possible without cluttering up the GUI. It indexes your code to give you relevant suggestions, and to complete your code for you. IntelliJ IDEA also automates several tasks that would otherwise be annoyingly repetitive. It's all that and a bag of chips, er, computer chips because it has an amazing out-of-the-box experience and doesn't just have to be used for Java. While it's focused on it, it's still great for SQL, JPQL, HTML, JavaScript, and other languages including when you inject a language into a string literal. You still get all the benefits of the IDE. 3. Aptana Studio 3 Aptana Studio 3 is one of the best IDE web development options available and it's also open source. You can customize the GUI, it has a built-in terminal, Git integration, a debugger as well as a deployment wizard. Autocomplete is also available for HTML, CSS, and JavaScript. 4. PyCharm PyCharm is primarily one of the best IDE for Python, but the premium version also supports other languages for web development. It checks for errors as you code and also has autocomplete. It has a sleek, minimal GUI for uncluttered development, and also checks the quality of your code to make sure it's up to snuff. 5. PhpStorm PhpStorm is another option to consider for one of the best IDE for web development. It's focused on PHP, but front-end languages are also fully supported including HTML 5, CSS, Sass, Less, JavaScript, and more. It also includes support for major software to create websites such as WordPress, Drupal, Joomla!, Laravek, and others. It's easy to navigate, has code autocomplete, debugging and testing tools as well as quick (and safe) refactoring. 6. WebStorm WebStorm is on the list as one of the best IDE for web development because it's excellent for JavaScript. It includes not only code autocompletion and live error-detection, but it also has a debugger, unit testing, and many integrations such as Git, GitHub, Mercurial, and others. It loads quickly on startup and can automate many tedious tasks for increased productivity. 7. NetBeans You can't have a list of the best IDE for web development without including NetBeans. It's among one of the most popular options for the best IDE because it's a no-nonsense software for Java, JavaScript, PHP, HTML 5, CSS, and more. It also helps you create bug-free websites by highlighting code not just syntactically, but also semantically. It also has a lot of powerful refactoring tools while being open source. 8. Eclipse This article also wouldn't be complete without mentioning Eclipse. It's an open-source and community-driven IDE for Javascript, mainly, but there are tools, for example, HTML, CSS. Its contributors include Google, Netflix, Facebook, GE, and Walmart. So, you can be sure you're getting one of the best IDE for web development. 9. RubyMine RubyMine is focused on Ruby and Ruby on Rails, but it's also tailored for JavaScript, CSS, Sass, Less, and more. It has syntax and error-highlighting, code completion, an advanced search for any class, file, or symbol, and fast navigation, to name just a few of the features. 10. Komodo IDE Komodo IDE is one of the best IDE for web development, but it's a premium option. It has a lot of features including autocomplete, visual debugger, unit testing, add-ons, team collaboration, project management tools, and more. It supports Git, Mercurial, Subversion, CVS, Perforce, and Bazaar as well as all programming languages. Which is the Best IDE for Web Development? As mentioned earlier, the answer to which is the best IDE for web development, it all comes down to your specific needs. Everyone's needs are different for their projects. Your pick for the best IDE may not be the same as one of your colleagues. That's okay. Sometimes, the smallest feature makes a world of difference. Best IDE Alternatives If you're looking at the list of features IDEs typically have and you're thinking, "That's just too much for me," there are other options available. Not only are there other IDEs out there, but there are also text and html editors that have some features in common with IDEs that are popular, quite powerful, and that you may find sufficient for your needs. Here's a short-list of the best IDE alternatives to check out: Visual Studio Code Vim Sublime Text 3 NotePad++ Komodo Edit Brackets In your opinion, which is the best IDE for web development? What's your pick for the best IDE for your specific needs? Still asking yourself, "What is an IDE?" Share your experience in the comments below. It's easy to get web developers and web designers mixed up – they both work on websites, use similar tools, and sometimes have similar skills. Or that's what it looks like from the outside looking in. In reality, designers and developers have very distinct roles and are responsible for different aspects of building a website. Here's an easy guide on how to tell the roles apart. Let's dig in. Table of Contents: Web Designer vs Web Developer: What's the Difference? Confused about the roles and responsibilities of web developers vs. web designers? You're not alone. Many still struggle to tell these two professions apart. While web designers and web developers work hand in hand to ship new websites to the world wide web, their roles are distinctly different. A great way to put these career paths into context is to think about the process of building a house. You need an architect to come up with a vision of the house – the visual elements, the floor plan and layout, materials, and the general look and feel of the space. And then you need construction workers to take these architectural drawings and execute them, turning a vision into a building. A web designer is like an architect. The designer's responsibilities involve developing a vision of how the website will look, feel, and function. And that's a broad set of tasks — from user research and information architecture to typography and user interface design. Web designers tend to specialize in certain areas: some focus on UX (user experience) design, others choose to work on UI (user interface) design, and there are some unicorn visual designers out there who can do both. A web developer will look after the code. Using various programming languages, such as HTML, CSS, or JavaScript, a web developer will build the website's framework and ensure it's bug-free and functions as expected. A web developer will also often act as an advisor to the designer, assessing whether the designs can be implemented technically and how complicated or time-consuming the development process will be. Since developers have to cover a wide range of responsibilities — programming the UI and maintaining the server-side of the website — the field is often split into different roles. You have front-end developers, back-end developers, and full-stack developers. Can you be both a web designer and a developer? Let's put it this way — countless people are great at both math and literature. Fundamentally, there isn't anything stopping you from mastering both ends of the skills spectrum. So the short answer is, yes, you can. The only catch to becoming a pro in both fields is that it might take you twice as long. What is Web Development? Web development is the process of building, creating, and maintaining websites and web applications. It's the work that turns a wireframe into a functioning website, ensuring it runs smoothly, loads fast, and delivers a seamless experience to the user. Who is a Web Developer? Development skills are some of the most sought-after skills in the market, making it a popular and, often, well-paid career path. That's mostly because of the rich and varied arsenal of skills a good developer will need to meet the job requirements. Web developers, or 'devs', juggle various coding languages to build and maintain websites. The programming languages they use will depend on the type of tasks they're performing. But generally, web development is broken down into two main categories: front-end development and back-end development, which refers to the server maintenance side. A full-stack developer can look after both front-end and back-end website needs and provide input on user experience and business logic. Web Development Roles Now let's look at what web developers in each role do. Front-End Developer A front-end developer is responsible for coding the mockup design for the website. Leveraging several programming languages (mainly HTML, CSS, and JavaScript), a front-end dev will implement the site's layout, visual elements, and interactivity. Must-Have Skills for Front-End Developer: Fluency in HTML, CSS, and JavaScript languages Knowledge of CSS and JavaScript frameworks and their key components Ability to leverage CSS preprocessors like Sass, LESS, or Stylus Handling of version control software like Git Using frameworks, such as React, to build powerful and modular applications Understanding responsive design principles Knowledge of browser developer tools Technical testing and debugging knowledge Understanding of web performance and automation tools, such as Grunt or gulp Knowledge of the command line (not compulsory but incredibly beneficial) Back-End Developer A back-end developer often referred to as 'the engineer', will work behind-the-scenes, writing and maintaining programs that power the frontend. This includes looking after the server that hosts the website, maintaining the application and the database. Back-end developers leverage a variety of server-side languages, such as PHP, Python, Ruby, and Java. Must-Have Skills for Back-End Developer Mastery of Python, which is often used as a server-side scripting language Knowledge of PHP, another scripting language used to request server content Knowledge of the Ruby on Rails software framework Mastery of SQL, the industry-standard language for querying and managing a database Deep understanding of API architectures, namely REST and SOAP technologies Understanding of version control systems like Git or GitHub Knowledge of designing backend systems and applications Handling of technical and functional documentation of systems and applications Full-Stack Developer A full-stack developer has sufficient knowledge in both front-end and back-end development. With a well-rounded understanding of how the web works on all levels, full-stackers make sure the website's front-end and back-end tie in and work smoothly together. Full-stack developers may also be expected to suggest and maintain the best practices, anticipate tech issues, and contribute to the overall development strategy. Must-Have Skills for Full-Stack Developers Deep understanding of front-end libraries and frameworks Proficiency in website optimization, such as improving site/page load times Ability to build a site's backend using programming languages like Ruby, Angular, or Bootstrap Handling of sysadmin tasks Ability to run SQL queries on databases Ability to manage application deployment, server upgrades, and server-side integrations Technologies and Tools Commonly Used by Web Developers Front-end developers create the on-screen reality. What starts as a mockup in design software has to be translated into a scrollable, clickable website, and that's exactly what a front-end dev does. The back-end ecosystem, on the other hand, is what powers the website. Think of it as the brain of a site that elegantly combines the server, the database, and the server-side applications to function in harmony. How do you keep up with the tech requirements? Pulling together a well-balanced toolkit that covers all aspects of web development is a great way to stay on top of your game: Web Developer Portfolio Examples Codepuffin – a colorful, brand-first web development portfolio that features several beautiful visual elements, including unique typography and bold colors. Patrick David – an example of a front-end developer's digital portfolio that fits onto a single page. Mees Verberne – a moody, unique digital portfolio that shows off front-end development and design skills. Now that we've covered web development let's move on to design. What is Web Design? Web design is the process of researching, planning, conceptualizing and arranging content or design elements online. Today, web design goes beyond just aesthetics and encompasses more complex aspects, such as user experience and information architecture. In fact, web design can impact your website's performance in search engines like Google, ultimately affecting your bottom line. Who is a Web Designer? A web designer is responsible for the look and feel of a website. The person is usually well-versed in color schemes, graphic design, typography, and information flow. Besides designing user interfaces, web designers are also responsible for researching and understanding the end user's needs and motivations. Equipped with this knowledge, they can create enjoyable user experiences that streamline conversions and keep visitors coming back. Web Design Roles Here's what web designers in different roles do. UI Designers A UI designer concerns himself with building clean and appealing user interface designs based on the research gathered by UX designers. UI designers are responsible for every touchpoint and interaction between a user and a device — be that a computer, a mobile app, or a car dashboard. Must-Have Skills for UI Designers In-depth knowledge of color theory, typography, and UI design patterns Strong understanding of design approaches, such as the Gestalt Principles Proficiency in at least one industry-standard design and prototyping tool Understanding of responsive design fundamentals Knowledge of how to design UI elements, such as buttons, icons, sliders, and scrollbars Ability to develop and establish a style guide to maintain consistency Ability to animate and design interactivity UX Designers While there's some overlap between UI and UX designers' roles, the latter is responsible for figuring out the needs of the site's users (and potential users), determining user stories, creating user personas, and developing prototypes, and carrying out user testing. A UX designer must also ensure that there's a logical flow on the website, that all interactions are intuitive and meet user needs. Must-Have Skills for UX Designers Ability to conduct user research Knowledge of creating user personas Ability to determine and design the information architecture Proficiency in designing user flows and wireframes Proficiency in industry-standard design and wireframing tools like Figma or Sketch Ability to create prototypes Understanding of user testing Visual Designers A visual designer role is best described as a hybrid role between a graphic designer and a UI designer, with a bunch of extra skills thrown into the mix. Visual designers typically have a strong understanding of graphic design and branding and are experts at visual messaging. Their primary responsibilities involve aspects like typography, layout, spacing, imagery, and color palettes. Must-Have Skills for Visual Designers Strong graphic design basics (balance, harmony, hierarchy, etc.) In-depth knowledge of color theory, typography, and design patterns Proficiency in design software, such as the Adobe Creative Cloud Ability to work off a brief and interpret information creatively Experience with transitions, animation, motion design, and dynamic interaction Ability to conceptualize brand elements Understanding of responsive design fundamentals Technologies and Tools Commonly Used by Web Designers A practical, well-balanced toolkit can be an excellent asset for web designers. The time you save using helpful tools can be invested in more important tasks that drive quality results. The problem is — web design tools are abundant in the market. Unless you know exactly what you need, it can be overwhelming to choose the best tool for the job. We've collected the most popular web design tools and software for your convenience. Web Designer Portfolio Examples The internet is full of thousands of digital portfolios with hand-picked work examples and case studies that help web designers land their next role or project. Here are a few stellar examples of web design portfolios: Elegant Seagulls – showing off highly polished, modern UI, this digital portfolio is a prime example of showing off your skills and taste as a designer. Tobias Ahlin – a clean, minimalistic digital portfolio of a product designer that consults tech companies and startups. Wokine – this Paris-based web design agency uses a split-screen design to show off their work and brand. Now let's move on to the money-part. Web Designer and Web Developer Salary Trends Who makes more money: web designers or web developers? In general, surveys show that web developers earn more than web designers, but it really depends on how you slice the data. Even among the different types of developer roles, there are significant differences in salaries. Your salary will depend more on the skills you bring to the table than your official job title. Web Designer Salaries Web designers earn a decent salary. While the pay can vary depending on the type of design role, location, and the seniority of the position, the average yearly salary of a web developer in the US is about $57k. It's worth pointing out that the available salary ranges differ significantly from source to source. Payscale: $51k/year, with a low of $35k and a high of $74k Glassdoor: $53k/year, with a low of $38k and a high of $74k Salary.com: $65k/year, with a low of $50k and a high of $77k ZipRecruiter: $60k/year, with a low of $21k and a high of $114k Looking at salary ranges for the different design roles, it's rather apparent that the hybrid UI/UX designers rake in the highest earnings. Here's the full salary breakdown by design role. UI designers tend to make between $44k and $128k per year, on average earning $79k/year. Glassdoor: $85k/year, with a low of $59k and a high of $128k Payscale: $64k/year, with a low of $44k and a high of $92k Indeed: $87k/year UX designers earn between $50k and $128k per year, on average getting $87k per year. Payscale: $74k/year, with a low of $108k Glassdoor: $85k/year, with a low of $59k and a high of $128k Indeed: $103k/year UI/UX designers have the best salaries, on average, earning $89k per year. Glassdoor: $85k/year, with a low of $59k and a high of $128k ZipRecruiter: $96k/year, with a low of $34k and a high of $150k Indeed: $87k/year to $103k/year Web Developer Salaries Although on average, web developers earn higher salaries than web designers, the pay differs based on their specialization. Developers tend to make more money than designers primarily because of their ability to write code in different languages. Coding is a highly valued skill in the modern, tech-driven world, but not all programming skills are valued equally. On average, a web developer in the US makes between $65k-$75k per year. Let's take a closer look at the average salaries for each role. Front-end developers in the US can expect to make, on average, $70-$75k per year. Payscale: $73k/year, with a low of $49k and a high of $113k Glassdoor: on average, $77k/year, with the top tier making up to $131k Indeed: $117k/year; however, this is definitely on the higher end of the range Back-end developers tend to earn a higher salary, with the aver1age pay hovering around the $100k per year mark. Glassdoor: $101k/year on average, with the lowest percent earning $81K and the top tier making $124k/year Payscale: $75k/year, surprisingly just a little more than frontend developer salaries listed on this site Indeed: a whopping $127k/year Full-stack developers' salaries vary significantly, possibly because there are fewer specialists with this range of skills. On average, a full-stack developer in the US can expect to earn around $90.5k per year, although the top tier rake in a lot more. Glassdoor: $68.5k/year on average, yet the top-paid developers get up to $111k/year Payscale: $75k/year, with a low of $48k and a high of $113k Indeed: $113.5k/year, with the less experienced specialists making around $92k/year Salary.com: slightly above $95k/year, with a range of $81k to $108k Naturally, the better you are at your job, the higher salary you can command. With technologies changing so fast and new programming languages emerging every year, web developers must continue learning and adding new skills to their portfolio to ensure an upward trajectory in their salaries. How to Learn Web Design and Web Development Now it's time to cover where and how to learn web design and web development. Get a Formal Degree or Get Certified If you like learning in a structured, offline environment, one of the more traditional routes of education could be the right option for you. Both design and web development learning programs have a lot to offer; you just need to know which profession you want to pursue. Consider the following paths: University – courses take between 2 to 4 years and you get to throw a hat in the air at the end of the course. Technical Schools – typically more hands-on, focused on developing practical skills faster, technical schools are great for professionals who know precisely what they're looking for. Boot Camps – intensive full-time & part-time learning programs aimed at equipping students with fundamental skills and knowledge. Besides the valuable training, you also get to connect with a like-minded community, find a mentor, and sometimes, if you're fortunate, get a job offer at the end of the course. The advantage of getting a formal education is that you graduate with a degree or an official certification, which can be quite the door opener when job hunting. Structured, teacher or instructor-led courses can help those who are starting from zero. The downside, of course, is the cost of traditional education as well as the restrictions of your location. If you're eager to get into a physical classroom, consider and research the options available in your area. Educate Yourself Some of the world's top web designers and web developers are self-taught. Learning on your own has many benefits: you get to learn in your own time and in a way that fits your schedule. Plus, with the Internet being a free resource, you can educate yourself without incurring any student loan debt. There are hundreds of free and paid design and development courses online that can teach you everything from design fundamentals to server-side coding languages. Where do you start? Here are some top recommended resources for a junior designer and developer: YouTube – there are millions of free videos and tutorials from highly experienced professionals that can give you a headstart in your learning journey. Try researching top names in your industry first to avoid having to sift through so many videos searching for the right fit. Once you identify the experts, see if they share any helpful content on YouTube, their website, or another platform. Tutorial Sites – with plenty of highly-focused, in-depth courses, tutorial sites can be an excellent stop for picking up technical skills and figuring out how to accomplish specific results in your code or design. One thing to watch out for, though, is content recency – as design software continues to evolve rapidly and new programming languages emerge all the time, you're running a risk of learning something that's irrelevant or, worse, incorrect. To get the latest, best quality web development and design tutorials and tips, join the WebsiteSetup newsletter and start learning as you go. To Sum Up: Should You Hire a Web Developer or Web Designer? Hopefully, this guide has provided you with insight into the key differences between a web developer and a web designer. So, let's recap on what we've covered so far. If you need someone to create visuals for your website, including the brand style, layout, information flow, and user interfaces, you'll be looking to hire a web designer. Like an architect, the web designer will develop a vision for your website: the look and feel, the stand-out elements, and the must-have functionalities for a seamless user experience. When you have the mockups and wireframes provided by the web designer, you'll need a front-end web developer to code them up. Using one of the front-end coding languages, the front-end developer will turn these designs into a functioning website – you'll have clickable, scrollable elements, site navigation, and content styling that matches your brand guidelines. You might then need the help of a back-end developer, who will take care of all server-side programming, third-party integrations, and programs that are necessary to power a website's server and database. Alternatively, you may look for a full-stack developer who understands both front-end and back-end processes and can build effective programs needed for a website. If it's a small project, they can probably write all of a site's code by themselves. And that's how you decide who to hire for your project – easy, right?