### Testování Cypress pro začátečníky

**Celková doba trvání:** 60 hodin  
**Počet lekcí**: 20  
**Délka jedné lekce:** 3 hodiny

### Modul 1: Úvod do automatizovaného testování, základy programování a nastavení Cypressu (lekce 1-5)

#### Lekce 1: Uvítání, přehled kurzu a úvod do Node.js

* **Cíle:**
  + Porozumět struktuře, cílům a očekáváním kurzu.
  + Získejte úvodní informace o Node.js a jeho úloze v systému Cypress.
* **Obsah:**
  + **Úvod do kurzu:**
    - Přehled automatizovaného a manuálního testování.
    - Výhody učení se Cypressu pro automatizaci testování.
  + **Úvod do Node.js:**
    - Co je Node.js?
    - Instalace Node.js a npm (Správce balíčků Node).
    - Základní koncepty Node.js relevantní pro Cypress.
  + **Nastavení vývojového prostředí:**
    - Instalace potřebných nástrojů (VS Code, Git).
    - Nastavení úložiště kódu (GitHub/GitLab).
* **Aktivity:**
  + Nainstalujte Node.js a ověřte instalaci.
  + Nastavte úložiště Git a naklonujte startovací projekt.
* **Zdroje:**
  + [Oficiální dokumentace Node.js](https://nodejs.org/en/docs/)
  + Instalační příručky pro vývojové nástroje.

#### Lekce 2: Základy jazyka JavaScript pro automatizaci testů

* **Cíle:**
  + Naučte se základní koncepty jazyka JavaScript potřebné pro psaní testů Cypress.
* **Obsah:**
  + **Základy JavaScriptu:**
    - Proměnné (var, let, const) a datové typy.
    - Operátory (aritmetické, porovnávací, logické).
  + **Základní syntaxe a kódovací konvence:**
    - Psaní a spouštění jednoduchých skriptů v jazyce JavaScript.
    - Porozumění souboru console.log a základům ladění.
* **Aktivity:**
  + Psaní jednoduchých skriptů v jazyce JavaScript k procvičení proměnných a operátorů.
  + Cvičení na tisk výstupu a základní výpočty.
* **Zdroje:**
  + [Výukové programy pro JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide)
  + Interaktivní kódovací platformy, jako je [Codecademy](https://www.codecademy.com/) nebo [FreeCodeCamp](https://www.freecodecamp.org/).

#### Lekce 3: Řídicí struktury a funkce v jazyce JavaScript

* **Cíle:**
  + Porozumět řídicím strukturám a funkcím v jazyce JavaScript a implementovat je.
* **Obsah:**
  + **Řídicí struktury:**
    - Podmíněné příkazy: if, else if, else.
    - Smyčky: for, while, do-while.
  + **Funkce:**
    - Deklarace a výrazy funkcí.
    - Parametry a návratové hodnoty.
    - Rozsah a zvedání.
* **Aktivity:**
  + Vytváření skriptů pomocí smyček a podmíněných příkazů.
  + Psaní a vyvolávání funkcí pro provádění konkrétních úloh.
* **Zdroje:**
  + Interaktivní cvičení kódování řídicích struktur a funkcí.
  + [Řídicí tok JavaScriptu](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Control_flow_and_error_handling).

#### Lekce 4: Pole, objekty, JSON a DOM

* **Cíle:**
  + Naučte se pracovat s poli a objekty, které jsou nezbytné pro práci s testovacími daty.
  + Porozumět objektovému modelu dokumentu (DOM) a jeho vlastnostem a metodám.
* **Obsah:**
  + **Pole:**
    - Vytváření prvků pole a přístup k nim.
    - Běžné metody pole (push, pop, shift, unshift, map, filter).
  + **Objekty:**
    - Vytváření objektů a přístup k vlastnostem.
    - Vnořené objekty a manipulace s objekty.
  + **Základy JSON:**
    - Porozumění struktuře JSON.
    - Parsování a řetězení dat JSON.
  + **Úvod do DOM:**
    - Co je DOM?
    - Vlastnosti a metody prvků DOM.
    - Navigace a manipulace s DOM pomocí JavaScriptu.
* **Aktivity:**
  + Manipulujte s poli a objekty prostřednictvím kódovacích cvičení.
  + Převod objektů JavaScript na JSON a naopak.
  + Prozkoumejte a manipulujte se systémem DOM na jednoduché stránce HTML.
* **Zdroje:**
  + [Pole JavaScriptu](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array)
  + [Objekty JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Working_with_Objects)
  + [Výukový program JSON](https://www.json.org/json-en.html)
  + [Úvod do DOM](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction)

#### Lekce 5: Úvod do DOM: Vlastnosti, metody a objekt globálního okna.

* **Cíle:**
  + Prohloubení znalostí o DOM a globálním objektu okna.
  + Naučte se klíčové vlastnosti a metody pro programovou interakci s prvky DOM.
* **Obsah:**
  + **Podrobné koncepty DOM:**
    - Hierarchická struktura DOM.
    - Přístup k prvkům pomocí metod jako getElementById, querySelector, getElementsByClassName.
  + **Vlastnosti prvků DOM:**
    - Běžné vlastnosti jako innerHTML, textContent, value, style.
    - Porozumění atributům prvků a manipulace s nimi (src, href, atributy data-\*).
  + **Metody prvků DOM:**
    - Manipulace s DOM pomocí metod jako appendChild, removeChild, replaceChild.
    - Zpracování událostí: Přidávání a odebírání posluchačů událostí.
  + **Globální okno Objekt:**
    - Přehled objektu okna a jeho vlastností.
    - Běžné metody okna: alert, confirm, prompt, setTimeout, setInterval.
    - Pochopení vztahu mezi oknem a dokumentem.
* **Aktivity:**
  + Procvičte si přístup k prvkům DOM a manipulaci s nimi pomocí jazyka JavaScript.
  + Vytvoření interaktivních prvků na jednoduché stránce HTML pomocí metod DOM.
  + Prozkoumejte objekt okna prostřednictvím cvičení na kódování.
* **Zdroje:**
  + [Průvodce manipulací DOM](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Manipulating_the_DOM)
  + [Dokumentace objektu okna](https://developer.mozilla.org/en-US/docs/Web/API/Window)

### Modul 2: Úvod do Cypressu a základní pojmy (lekce 6-9)

#### Lekce 6: Instalace a konfigurace aplikace Cypress

* **Cíle:**
  + Nastavte Cypress ve vývojovém prostředí a pochopte jeho strukturu.
* **Obsah:**
  + **Instalace Cypress:**
    - Použití npm k instalaci Cypressu.
    - Ověření instalace.
  + **Struktura složky Cypress:**
    - Porozumění výchozím adresářům (svítidla, integrace, zásuvné moduly atd.).
  + **Nastavení konfigurace:**
    - Přehled souboru cypress.config.js.
    - Nastavení baseUrl a dalších globálních konfigurací.
  + **Psaní prvního testu Cypress:**
    - Vytvoření jednoduchého testu pro návštěvu webové stránky a kontrolu jejího názvu.
* **Aktivity:**
  + Nainstalujte Cypress a spusťte výchozí ukázkové testy.
  + Napište a proveďte základní test Cypress.
* **Zdroje:**
  + [Instalační příručka Cypress](https://docs.cypress.io/guides/getting-started/installing-cypress)
  + [Struktura složky Cypress](https://docs.cypress.io/guides/core-concepts/writing-and-organizing-tests#Folder-Structure)

#### Lekce 7: Interakce s webovými prvky a zpracování uživatelských vstupů

* **Cíle:**
  + Získejte praktické zkušenosti s interakcí s webovými prvky a zpracováním uživatelských vstupů pomocí aplikace Cypress.
* **Obsah:**
  + **Interakce s webovými prvky:**
    - Příkazy pro kliknutí, psaní, výběr a najetí (cy.click(), cy.type(), cy.select(), cy.hover()).
    - Obsluha rozevíracích seznamů, zaškrtávacích políček, přepínačů a tlačítek.
  + **Zpracování uživatelských vstupů:**
    - Správa odesílání formulářů.
    - Ověřování vstupních polí uživatele.
    - Simulace interakcí uživatele (přetahování, události na klávesnici).
* **Aktivity:**
  + Napište testy, které provedou různé interakce s ukázkovou webovou aplikací.
  + Simulujte scénáře uživatelských vstupů a ověřujte výsledky.
* **Zdroje:**
  + [Dokumentace příkazů Cypress](https://docs.cypress.io/api/table-of-contents)
  + Příklady projektů s interaktivními formuláři.

#### Lekce 8: Práce s tvrzeními a ověřeními

* **Cíle:**
  + Naučte se psát efektivní tvrzení pro ověření výsledků testů.
* **Obsah:**
  + **Tvrzení s Chai:**
    - Úvod do knihovny Chai assertion používané společností Cypress.
    - Typy tvrzení: should, expect a assert.
    - Psaní jednoduchých a složených tvrzení.
  + **Běžné typy tvrzení:**
    - Kontrola existence, viditelnosti, rovnosti, délky a obsahu.
  + **Osvědčené postupy:**
    - Jasná a popisná tvrzení.
    - Vyhnutí se chybným testům pomocí spolehlivých tvrzení.
* **Aktivity:**
  + Psaní testů s různými typy tvrzení.
  + Přepracování testů s cílem zlepšit srozumitelnost a spolehlivost tvrzení.
* **Zdroje:**
  + [Knihovna tvrzení Chai](https://www.chaijs.com/)
  + [Průvodce tvrzeními Cypress](https://docs.cypress.io/guides/references/assertions)

#### Lekce 9: Práce s přípravky a správa testovacích dat

* **Cíle:**
  + Efektivní správa testovacích dat pomocí fixtures a porozumění vzorům zahrnutí/vyloučení.
* **Obsah:**
  + **Používání příslušenství:**
    - Vytváření souborů s přípravky (JSON).
    - Načítání dat z přípravku v testech (cy.fixture()).
    - Strukturování přípravků pro opakované použití.
  + **Zahrnout/vynechat vzory:**
    - uspořádání testů tak, aby zahrnovaly nebo vylučovaly určité sady testů.
    - Používání značek nebo konvencí pro pojmenování pro správu testů.
  + **Dynamické zpracování dat:**
    - Generování náhodných dat pro testy.
    - Parametrizace testů s různými soubory dat.
* **Aktivity:**
  + Vytváření a používání souborů přípravků v testech Cypress.
  + Implementace vzorů zahrnutí/vyloučení pro správu provádění testů.
* **Zdroje:**
  + [Dokumentace svítidel Cypress](https://docs.cypress.io/api/commands/fixture)
  + Ukázkové soubory přípravků a příklady testovacích dat.

### Modul 3: Strukturování a organizace testů Cypress (lekce 10-14)

#### Lekce 10: Strukturování testovacích sad a testovacích případů

* **Cíle:**
  + Organizujte testy pomocí osvědčených postupů pro udržovatelnost a škálovatelnost.
* **Obsah:**
  + **Organizace testů:**
    - Použití describe a it bloků ke strukturování testovacích sad a testovacích případů.
    - Seskupování souvisejících testů.
  + **Metody nastavení a demontáže:**
    - Použití háčků before, beforeEach, after a afterEach.
    - Nastavení předběžných podmínek a úklid po testech.
  + **Označování a kategorizace testů:**
    - Použití značek ke kategorizaci testů pro selektivní provádění.
    - Zavedení konvencí pojmenování pro přehlednost.
* **Aktivity:**
  + Strukturovat sadu testů pomocí describe a it.
  + Implementace háčků pro správu nastavení a ukončení testů.
* **Zdroje:**
  + [Testovací struktura Cypress](https://docs.cypress.io/guides/core-concepts/writing-and-organizing-tests#Structure)
  + Příklady sad testů s uspořádanými strukturami.

#### Lekce 11: Vzor objektového modelu stránky (POM) v Cypressu

* **Cíle:**
  + Implementace vzoru POM (Page Object Model) pro zvýšení udržovatelnosti testů.
* **Obsah:**
  + **Porozumění POM:**
    - Co je to objektový model stránky?
    - Výhody použití POM při automatizaci testů.
  + **Implementace POM:**
    - Vytváření tříd/modulů stránek.
    - Zapouzdření prvků a akcí stránky do objektů stránky.
    - Opakované použití objektů stránky ve více testech.
  + **Osvědčené postupy:**
    - Udržování čistoty a zaměření objektů na stránce.
    - Zamezení duplikace a zajištění škálovatelnosti.
* **Aktivity:**
  + Přepracujte stávající testy tak, aby používaly vzor POM.
  + Vytvoření objektů stránek pro různé části ukázkové aplikace.
* **Zdroje:**
  + [Objektový model stránky v Cypressu](https://docs.cypress.io/guides/references/best-practices#Page-Object-Model)
  + Ukázkové projekty demonstrující POM.

#### Lekce 12: Konfigurační soubory a proměnné prostředí Cypress

* **Cíle:**
  + Porozumět konfiguračním souborům Cypress a proměnným prostředí pro flexibilní nastavení testů a využívat je.
* **Obsah:**
  + **Konfigurační soubory Cypress:**
    - Přehled souboru cypress.config.js.
    - Nastavení globálních konfigurací (např. baseUrl, nastavení viewport).
    - Přizpůsobení chování Cypressu pomocí konfiguračních souborů.
  + **Proměnné prostředí:**
    - Používání proměnných prostředí pro citlivé údaje (např. klíče API, adresy URL).
    - Správa různých prostředí (vývojové, staging, produkční).
    - Přístup k proměnným prostředí v testech (Cypress.env()).
  + **Osvědčené postupy:**
    - Udržování uspořádané a bezpečné konfigurace.
    - Vyhnout se pevně zadaným hodnotám v testech.
* **Aktivity:**
  + Konfigurace nastavení Cypressu pomocí souboru cypress.config.js.
  + Implementace proměnných prostředí pro různé testovací scénáře.
* **Zdroje:**
  + [Konfigurační příručka Cypress](https://docs.cypress.io/guides/references/configuration)
  + Příklady použití proměnných prostředí v Cypressu.

#### Lekce 13: Praktické použití jazyka TypeScript s aplikací Cypress

* **Cíle:**
  + Zavedení jazyka TypeScript do testů Cypress s cílem zvýšit kvalitu kódu a zlepšit zkušenosti vývojářů.
* **Obsah:**
  + **Úvod do jazyka TypeScript:**
    - Co je TypeScript?
    - Výhody použití jazyka TypeScript s aplikací Cypress.
  + **Nastavení jazyka TypeScript v jazyce Cypress:**
    - Instalace jazyka TypeScript a potřebných typů.
    - Konfigurace souboru tsconfig.json pro Cypress.
  + **Psaní testů Cypress v jazyce TypeScript:**
    - Převod testů JavaScriptu do jazyka TypeScript.
    - Použití typových anotací a rozhraní.
    - Využití technologie IntelliSense a kontroly typu.
  + **Osvědčené postupy:**
    - Uspořádání souborů TypeScript.
    - Zachování bezpečnosti typu při testech.
* **Aktivity:**
  + Nastavení jazyka TypeScript v projektu Cypress.
  + Napište a spusťte jednoduchý test Cypress pomocí jazyka TypeScript.
* **Zdroje:**
  + [TypeScript s Cypressem](https://docs.cypress.io/guides/tooling/typescript-support)
  + [Dokumentace jazyka TypeScript](https://www.typescriptlang.org/docs/)

#### Lekce 14: Používání baseUrl a kontextů v Cypressu

* **Cíle:**
  + Využití konfigurací baseUrl a kontextu pro zjednodušení psaní a provádění testů.
* **Obsah:**
  + **baseUrl:**
    - Nastavení baseUrl v konfiguraci Cypress.
    - Výhody použití baseUrl pro relativní adresy URL v testech.
    - Přepisování baseUrl pro různá prostředí.
  + **Souvislosti:**
    - Porozumění kontextům Cypress pro organizaci testů.
    - Použití kontextů pro správu různých testovacích scénářů.
  + **Osvědčené postupy:**
    - Konzistentní používání baseUrl napříč testy.
    - Strukturování kontextů pro přehlednost a udržovatelnost.
* **Aktivity:**
  + Konfigurace baseUrl a aktualizace stávajících testů tak, aby používaly relativní adresy URL.
  + Uspořádejte testy do různých kontextů pro různé scénáře.
* **Zdroje:**
  + [Konfigurace Cypress baseUrl](https://docs.cypress.io/guides/references/configuration#Base-Url)
  + Příklady použití kontextů v Cypressu.

### Modul 4: Pokročilé funkce Cypressu a praktické aplikace (lekce 15-20)

#### Lekce 15: Síťové požadavky a základní testování API pomocí Cypressu

* **Cíle:**
  + Zjistěte, jak zachytávat a zesměšňovat síťové požadavky a provádět základní testování API.
* **Obsah:**
  + **Přijímání síťových požadavků:**
    - Použití funkce cy.intercept() k zachycení požadavků HTTP.
    - Odpovědi pro konzistentní testovací prostředí.
  + **Základní testování API:**
    - Odesílání požadavků API v rámci testů Cypress.
    - Ověřování odpovědí API.
    - Kombinace testů uživatelského rozhraní a API pro komplexní pokrytí.
  + **Osvědčené postupy:**
    - Izolování testů pomocí zesměšňování síťových odpovědí.
    - Zajištění deterministických a spolehlivých testů.
* **Aktivity:**
  + Zachycení a vyřazení síťových požadavků v testu Cypress.
  + Napište základní test API pomocí Cypressu.
* **Zdroje:**
  + [Požadavky na síť Cypress](https://docs.cypress.io/api/commands/intercept)
  + Příklady testování rozhraní API pomocí společnosti Cypress.

#### Lekce 16: Úvod do zásuvných modulů a rozšíření Cypressu (zkráceně)

* **Cíle:**
  + Prozkoumejte zásuvné moduly a rozšíření Cypressu, které rozšiřují možnosti testování.
* **Obsah:**
  + **Zásuvné moduly Cypress:**
    - Co jsou zásuvné moduly Cypress?
    - Instalace a konfigurace základních zásuvných modulů Cypress.
    - Přehled oblíbených zásuvných modulů (např. cypress-axe pro testování přístupnosti).
  + **Rozšíření:**
    - Rozšíření funkcí Cypressu pomocí rozšíření.
    - Vlastní zásuvné moduly pro specifické potřeby testování.
  + **Osvědčené postupy:**
    - Výběr vhodných zásuvných modulů, aby nedošlo k přeplnění testovací sady.
    - Údržba konfigurací zásuvných modulů.
* **Aktivity:**
  + Instalace a konfigurace zásuvného modulu Cypress v ukázkovém projektu.
  + Využití zásuvného modulu k provedení dalšího typu testu (např. přístupnosti).
* **Zdroje:**
  + [Dokumentace zásuvných modulů Cypress](https://docs.cypress.io/plugins/)
  + Příklady zásuvných modulů Cypress a jejich použití.

#### Lekce 17: Problémy a řešení v reálném světě - příklady z praxe

* **Cíle:**
  + Řešení běžných problémů, s nimiž se setkáváte při používání softwaru Cypress v reálných scénářích.
  + Poskytnout praktická řešení a osvědčené postupy k překonání těchto problémů.
* **Obsah:**
  + **Běžné problémy s cypřiši:**
    - Zpracování dynamického obsahu a asynchronního chování.
    - Řešení vadných testů a zajištění spolehlivosti testů.
    - Správa složitých interakcí s uživateli a stavových aplikací.
  + **Praktická řešení:**
    - Implementace efektivních čekacích strategií (cy.wait(), asertions pro čekání na prvky).
    - Použití opakovaných pokusů a inteligentních selektorů ke snížení vločkovitosti.
    - Strategie pro správu stavu aplikace a závislostí v testech.
  + **Osvědčené postupy:**
    - Psaní udržovatelných a škálovatelných testů.
    - Uspořádání testovacího kódu tak, aby byl čitelný a opakovaně použitelný.
    - Využití funkcí Cypress k zefektivnění vývoje testů.
  + **Případové studie a příklady:**
    - Reálné příklady testů Cypress, které řeší konkrétní problémy.
    - Postupné ukázky řešení problémů.
* **Aktivity:**
  + Analyzujte a odstraňujte problémy s ukázkovými chybovými testy.
  + Přepracování problematických testů pomocí osvědčených postupů.
  + Vývoj řešení složitých scénářů interakce v ukázkové aplikaci.
* **Zdroje:**
  + [Osvědčené postupy společnosti Cypress](https://docs.cypress.io/guides/references/best-practices)
  + Příklady případových studií a návody na řešení problémů.

#### Lekce 18: Cypress Dashboard, Reporting a Continuous Integration and Delivery (CI/CD)

* **Cíle:**
  + Porozumět výhodám a funkcím Cypress Dashboard.
  + Naučte se vytvářet a interpretovat zprávy o testech.
  + Integrace testů Cypress s CI/CD pipelines pro automatizované testovací pracovní postupy.
* **Obsah:**
  + **Cypress Dashboard:**
    - Přehled funkcí Cypress Dashboard.
    - Výhody používání Cypress Dashboard pro správu testů a reportování.
    - Nastavení integrace s nástrojem Cypress Dashboard.
  + **Podávání zpráv:**
    - Generování testovacích zpráv pomocí aplikace Cypress.
    - Přizpůsobení výstupů sestavy.
    - Analýza výsledků testů a metrik.
  + **Continuous Integration (CI) a Continuous Delivery (CD):**
    - Úvod do konceptů CI/CD.
    - Přehled populárních nástrojů CI/CD (např. GitHub Actions, GitLab CI, Jenkins).
    - Konfigurace testů Cypress pro běh v potrubí CI/CD.
    - Automatizace provádění testů při revizích a nasazeních kódu.
    - Nastavení oznámení a upozornění na základě výsledků testů.
  + **Osvědčené postupy:**
    - Udržování testovacích prostředí v rámci CI/CD pipelines.
    - Zajištění bezpečné práce s proměnnými prostředí a tajemstvími v CI/CD.
* **Aktivity:**
  + Nastavte Cypress Dashboard pomocí ukázkového projektu.
  + Nakonfigurujte CI pipeline (např. pomocí GitHub Actions) pro automatické spouštění testů Cypress.
  + Vytvářet a interpretovat zprávy o testech.
* **Zdroje:**
  + [Cypress Dashboard Dokumentace](https://docs.cypress.io/guides/dashboard/introduction)
  + [Průvodci integrací nástrojů CI pro Cypress](https://docs.cypress.io/guides/continuous-integration/introduction)
  + [Zásuvné moduly Cypress Reporting](https://docs.cypress.io/guides/tooling/plugins-guide)

#### Lekce 19: Stručný přehled pokročilých témat

* **Cíle:**
  + Představte pokročilá témata Cypressu, abyste vzbudili zájem, aniž byste studenty zahltili.
* **Obsah:**
  + **Přehled pokročilých témat:**
    - Úplné testování rozhraní API s firmou Cypress.
    - Pokročilé integrace CI/CD.
    - Základy testování výkonnosti.
  + **Proč jsou tato témata důležitá:**
    - Zlepšení pokrytí a spolehlivosti testů.
    - Zjednodušení pracovních postupů vývoje.
    - Zajištění výkonu aplikace v souladu se standardy.
  + **Zdroje pro další vzdělávání:**
    - Oficiální dokumentace Cypress k pokročilým tématům.
    - Doporučené výukové programy a kurzy pro hloubkové ponory.
* **Aktivity:**
  + Diskutujte o scénářích, ve kterých by byla pokročilá témata přínosná.
  + Poskytnout návod, jak pokračovat ve studiu těchto témat po skončení kurzu.
* **Zdroje:**
  + [Cypress Advanced Guides](https://docs.cypress.io/guides/overview/why-cypress)
  + Odkazy na externí výukové programy a zdroje.

#### Lekce 20: Projekt Capstone, zhodnocení kurzu a další kroky

* **Cíle:**
  + Použít všechny naučené koncepty v komplexním projektu.
  + Zkontrolujte klíčový obsah kurzu a prodiskutujte další kroky pro pokračování ve vzdělávání.
* **Obsah:**
  + **Projekt Capstone:**
    - Vytvoření komplexní sady testů Cypress pro ukázkovou webovou aplikaci.
    - Implementace objektového modelu stránky, fixtures, proměnných prostředí a tvrzení.
    - Integrace Cypress Dashboard a CI/CD pipeline.
  + **Přehled kurzů:**
    - Rekapitulace klíčových pojmů a lekcí.
    - Řešení společných problémů a řešení.
  + **Další kroky:**
    - Další zkoumání pokročilých témat (např. kompletní testování API, pokročilé integrace CI/CD).
    - Zdroje pro další vzdělávání a profesní rozvoj.
  + **Zpětná vazba a otázky a odpovědi:**
    - shromažďování zpětné vazby od studentů ke kurzu.
    - Otevřené setkání s otázkami a odpověďmi, kde se zodpoví všechny zbývající otázky.
* **Aktivity:**
  + Dokončit a prezentovat hlavní projekt.
  + Zúčastněte se hodnocení a diskuse v rámci celého kurzu.
* **Zdroje:**
  + [Příklady Cypress](https://github.com/cypress-io/cypress-example-recipes)
  + Další vzdělávací zdroje a komunity.

### Další vylepšení kurzu

1. **Praktická cvičení:**
   * Každá lekce obsahuje praktická cvičení, v nichž studenti píší a spouštějí testy Cypress, čímž si upevňují znalosti a posilují sebedůvěru.
2. **Zadání a hodnocení:**
   * Týdenní úkoly k posouzení porozumění a poskytnutí zpětné vazby.
   * Kvízy nebo miniprojekty na konci každého modulu pro zjištění pokroku.
3. **Zdroje a odkazy:**
   * Poskytněte studentům doplňkové materiály, jako jsou odkazy na dokumentaci, přehledy a doporučená literatura, které jim pomohou při učení.
4. **Podpora a komunita:**
   * Podporujte vytváření studijních skupin nebo fór, kde mohou studenti klást otázky, sdílet znalosti a spolupracovat na projektech.
5. **Integrace jazyka TypeScript:**
   * Ačkoli je tato možnost představena v lekci 13, je použití jazyka TypeScript nepovinné, ale doporučuje se těm, kteří mají zájem o zvýšení kvality svého kódu.
6. **Úvod do pokročilých témat:**
   * V pozdějších lekcích se krátce zmiňte o pokročilých tématech, jako je testování celého API, Cypress Cloud a integrace CI, abyste vzbudili zájem a studenty nezahltili.

### Závěr

Po úspěšném absolvování kurzu **Cypress Testování pro začátečníky** budou vaši studenti dobře vybaveni základními dovednostmi a praktickými zkušenostmi potřebnými pro přechod na pozice automatizovaného testování. Budou schopni navrhovat, implementovat a udržovat sady automatizovaných testů pomocí Cypressu, integrovat procesy testování do pracovních postupů vývoje a přispívat k celkové kvalitě a spolehlivosti softwarových aplikací.

### Kompetence

Po absolvování kurzu si studenti osvojí následující kompetence:

1. **Základní programátorské dovednosti:**
   * **Znalost jazyka JavaScript:**
     + Znalost základů jazyka JavaScript, včetně proměnných, datových typů, operátorů, řídicích struktur (cyklů a podmínek), funkcí, polí a objektů.
   * **Manipulace s DOM:**
     + Schopnost interakce s objektovým modelem dokumentu (DOM) a manipulace s ním pomocí jazyka JavaScript.
   * **Asynchronní programování:**
     + Znalost asynchronních konceptů jazyka JavaScript, včetně zpětných volání, příkazů Promises a async/await.
2. **Odborné znalosti systému Cypress Framework:**
   * **Instalace a konfigurace:**
     + Znalost instalace Cypressu, nastavení vývojového prostředí a konfigurace nastavení Cypressu (cypress.config.js).
   * **Psaní a organizace testů:**
     + Schopnost psát strukturované a udržovatelné sady testů a testovací případy pomocí describe a it bloků.
     + Implementace vzoru POM (Page Object Model) pro lepší organizaci a opakované použití testů.
   * **Interakce s webovými prvky:**
     + Znalost používání příkazů Cypress pro interakci s různými webovými prvky (klikání, psaní, výběr atd.) a zpracování uživatelských vstupů.
   * **Tvrzení a ověření:**
     + Dovednost psát efektivní tvrzení pomocí Chai k ověření výsledků testů a zajištění spolehlivosti aplikace.
   * **Správa testovacích dat:**
     + Schopnost spravovat testovací data pomocí fixtures, implementovat vzory include/exclude a pracovat s dynamickými daty v testech.
   * **Správa životního prostředí:**
     + Porozumění proměnným prostředí a jejich použití pro různé scénáře testování.
3. **Pokročilé testovací techniky:**
   * **Základní testování API:**
     + Schopnost provádět základní testování rozhraní API zachycením síťových požadavků, jejich testováním a ověřováním odpovědí rozhraní API.
   * **Cypress Dashboard a reportování:**
     + Znalost nástroje Cypress Dashboard pro správu testů, podávání zpráv a analýzu výsledků testů.
   * **Kontinuální integrace a doručování (CI/CD):**
     + Znalost integrace testů Cypress v rámci CI/CD pipeline pro automatizaci testovacích pracovních postupů.
   * **Řešení problémů v reálném světě:**
     + Schopnost identifikovat a řešit běžné problémy související s testováním v systému Cypress pomocí praktických technik řešení problémů.
4. **Další technické dovednosti:**
   * **Integrace jazyka TypeScript:**
     + Základní znalost používání jazyka TypeScript s jazykem Cypress pro zvýšení kvality a udržovatelnosti kódu (nepovinné, ale doporučené).
   * **Zásuvné moduly a rozšíření Cypress:**
     + Znalost základních zásuvných modulů a rozšíření Cypress pro rozšíření možností testování.
   * **Osvědčené postupy v automatizaci testování:**
     + Znalost osvědčených postupů pro psaní udržovatelných, škálovatelných a spolehlivých automatizovaných testů.
5. **Měkké dovednosti:**
   * **Analytické myšlení:**
     + Schopnost analyzovat chování aplikace a navrhovat efektivní testovací případy.
   * **Pozornost věnovaná detailům:**
     + Ostrý zrak pro identifikaci potenciálních problémů a zajištění důkladného pokrytí testů.
   * **Spolupráce a komunikace:**
     + Dovednost spolupracovat s vývojovými týmy a efektivně sdělovat výsledky testů a problémy.

### Pracovní povinnosti

Díky těmto kompetencím mohou absolventi na svých současných nebo budoucích pozicích převzít následující pracovní povinnosti:

1. **Automatizovaný vývoj testů:**
   * **Návrh a implementace automatizovaných testů:**
     + Vytvářet, psát a udržovat automatizované testovací skripty pomocí frameworku Cypress, aby se zajistil soulad funkčnosti aplikace s požadavky.
   * **Vývoj testovacích sad a případů:**
     + Organizovat a strukturovat sady testů a jednotlivé testovací případy pro komplexní pokrytí funkcí aplikace.
2. **Provádění a údržba testů:**
   * **Spouštění a monitorování testů:**
     + Provádění automatizovaných testů, sledování jejich výkonu a interpretace výsledků testů s cílem identifikovat chyby.
   * **Udržování testovacích skriptů:**
     + Aktualizace a refaktorizace stávajících testovacích skriptů s ohledem na změny funkcí aplikace nebo požadavků na testování.
3. **Správa testovacích dat:**
   * **Správa a využití testovacích dat:**
     + Vytvářejte a spravujte soubory fixture, zpracovávejte dynamická data v testech a zajistěte, aby testy byly datově řízené pro flexibilitu a škálovatelnost.
4. **Identifikace a hlášení chyb:**
   * **Zjišťování a dokumentace závad:**
     + Identifikovat chyby nebo problémy prostřednictvím automatizovaného testování, jasně je dokumentovat a spolupracovat s vývojovými týmy na včasném řešení.
5. **Integrace s vývojovými procesy:**
   * **Integrace CI/CD:**
     + Integrace automatizovaných testů v rámci CI/CD pipelines, aby bylo možné provádět průběžné testování a zajistit, že nové změny kódu nezpůsobí regrese.
   * **Využijte Cypress Dashboard:**
     + Pomocí nástroje Cypress Dashboard můžete spravovat průběh testů, analyzovat výsledky testů a vytvářet zprávy, které informují zúčastněné strany o stavu kvality aplikace.
6. **Spolupráce a komunikace:**
   * **Spolupráce s multifunkčními týmy:**
     + Spolupráce s vývojáři, inženýry QA a produktovými manažery s cílem porozumět požadavkům na testování a zajistit komplexní pokrytí testů.
   * **Poskytování zpětné vazby a doporučení:**
     + Na základě výsledků testů nabízejte poznatky a návrhy na zlepšení kvality aplikací a testovacích procesů.
7. **Řešení problémů a optimalizace:**
   * **Řešení problémů s testováním:**
     + Řešení běžných problémů v Cypressu, jako je zpracování asynchronního chování, snížení vachrlatosti testů a správa složitých interakcí s uživateli.
   * **Optimalizace výkonu testů:**
     + Implementovat strategie pro zvýšení rychlosti a spolehlivosti provádění testů a zajistit efektivní využití testovacích zdrojů.
8. **Dodržování osvědčených postupů:**
   * **Zavedení osvědčených postupů:**
     + Dodržování osvědčených postupů v oblasti automatizace testů pro vytváření udržovatelných a škálovatelných sad testů, které zajišťují dlouhodobou účinnost testů.
9. **Neustálé učení a zlepšování:**
   * **Aktualizujte si trendy v testování:**
     + Neustále se seznamujte s novými funkcemi, zásuvnými moduly a metodikami testování Cypress, abyste rozšířili možnosti automatizovaného testování.
   * **Podílejte se na sdílení znalostí:**
     + Sdílet znalosti a osvědčené postupy se členy týmu a podporovat tak kulturu neustálého zlepšování v oblasti automatizace testování.

### Potenciální pracovní role pro absolventy

Díky těmto kompetencím a odpovědnosti mohou absolventi vykonávat různé funkce v oblasti zajištění kvality softwaru a automatizace testování, mimo jiné:

* **Juniorní tester automatizace**
* **Inženýr automatizace QA**
* **Vývojář automatizace testování**
* **Analytik zajištění kvality softwaru (QA)**
* **SDET (Software Development Engineer in Test) - vstupní úroveň**
* **Inženýr QA**

### Akční kroky pro lektora pro jeho studenty

Abyste zajistili úspěšné zavedení a efektivitu kurzu **Cypress Testování pro začátečníky**, uvádíme několik dalších doporučení:

1. **Vypracování podrobných plánů výuky:**
   * **Cíle a výstupy:** Jasně definujte, čeho má každá lekce dosáhnout a jaké dovednosti si žáci osvojí.
   * **Materiály a zdroje:** Připravte si předem diapozitivy, ukázky kódu a referenční materiály.
   * **Pokyny krok za krokem:** Rozdělte každou lekci na zvládnutelné části s jasnými pokyny.
   * **Cvičení a úkoly:** Navrhněte praktická cvičení, která posílí pojmy vyučované v každé lekci.
   * **Hodnocení:** Vytvářejte kvízy nebo miniprojekty, abyste vyhodnotili porozumění a pokrok studentů.
2. **Připravte si praktické projekty:**
   * **Projekt Capstone:** Zajistěte, aby byl hlavní projekt komplexní a umožnil studentům uplatnit všechny dovednosti, které se naučili.
   * **Přírůstkové projekty:** V průběhu kurzu zavádějte menší projekty, abyste si vybudovali jistotu a dovednost.
3. **Vytvoření podpůrného systému:**
   * **Úřední hodiny:** Nabídněte pravidelné úřední hodiny, kde mohou studenti požádat o pomoc a vysvětlení.
   * **Diskusní fóra:** Vytvořte prostor (například kanál na Slacku nebo speciální fórum), kde mohou studenti spolupracovat, klást otázky a sdílet své postřehy.
   * **Odborné recenze:** Povzbuzujte studenty, aby si navzájem hodnotili své práce, a podpořte tak spolupráci ve výuce.
4. **Shromážděte a zapracujte zpětnou vazbu:**
   * **Průzkumy a formuláře zpětné vazby:** Pravidelně sbírejte zpětnou vazbu od studentů, abyste zjistili, co funguje dobře a co je třeba zlepšit.
   * **Iterujte a zlepšujte:** Využijte zpětnou vazbu k potřebným úpravám obsahu kurzu, tempa nebo metod výuky.
5. **Poskytnutí dalších zdrojů:**
   * **Odkazy na dokumentaci:** Sdílejte odkazy na oficiální dokumentaci Cypressu a další relevantní zdroje.
   * **Doporučená četba:** Navrhněte knihy, články nebo výukové programy pro studenty, kteří se chtějí hlouběji seznámit s určitými tématy.
   * **Kontrolní listy:** Vytvořte si praktické referenční příručky pro rychlý přístup k příkazům, osvědčeným postupům a běžným vzorům.
6. **Podporujte neustálé učení:**
   * **Kurzy pro pokročilé:** Nabídněte nebo doporučte navazující kurzy pro studenty, kteří mají zájem o pokročilá témata Cypressu nebo jiných automatizačních rámců.
   * **Certifikace:** Informujte studenty o certifikacích, které mohou posílit jejich pověření v oblasti automatizace testování.
7. **Úspěšné příběhy:**
   * **Projekty absolventů:** Vyzdvihněte úspěšné projekty nebo svědectví bývalých studentů, abyste motivovali současné účastníky.
   * **Podpora při zprostředkování zaměstnání:** Poskytování poradenství při tvorbě životopisu, přípravě na pohovor a strategiích hledání zaměstnání přizpůsobených pro role v oblasti automatizace testování.

### Závěrečné myšlenky

Poskytnete-li svým studentům správné znalosti a praktické dovednosti v oblasti testování Cypress, výrazně tím zvýšíte jejich kariérní vyhlídky ve stále se rozvíjející oblasti zajištění kvality softwaru a automatizace testování. Podporou podpůrného a poutavého vzdělávacího prostředí jim pomůžete plynule přejít z rolí manuálního testování na automatizované a vybavíte je nástroji, které potřebují k tomu, aby uspěli na svých současných i budoucích pozicích.

Pokud máte další otázky nebo potřebujete další pomoc při vývoji a realizaci kurzu, neváhejte se na nás obrátit. Přejeme vám i vašim studentům mnoho úspěchů!