

Pytania na Narzędzia Programistyczne

~~REGEX

1. Czym jest regex?

(Wyrażenie regularne) wzorec, który opisuje łańcuch symboli.

2. Napisz REGEX na kod pocztowy

`^[0-9]{2}-[0-9]{3}$` ok

3. Napisz REGEX na adres e-mail

`^[a-zA-Z0-9_]{2,48}@[a-zA-Z0-9]{2,16}\.[a-zA-Z]{1-6}$` ok

4. Co oznacza kwantyfikator "?" w wyrażeniach regularnych?

Zero lub jedno wystąpienie.

5. Co oznacza kwantyfikator "*" w wyrażeniach regularnych?

Zero lub więcej wystąpień.

6. Co oznacza kwantyfikator "+" w wyrażeniach regularnych?

Jedno lub więcej wystąpień.

7. Co oznacza {n} w wyrażeniach regularnych?

Dokładnie n-razy wystąpień.

8. Co oznacza [abc] w wyrażeniach regularnych?

Dowolny znak ze wskazanego zbioru

9. Co oznacza [^abc] w wyrażeniach regularnych?

Dowolny znak różny od tych w zbiorze

10. Co oznacza \w w wyrażeniach regularnych?

Dowolny znak typu: litera, cyfra, podkreślnik

11. Co oznacza \W w wyrażeniach regularnych?

Dowolny znak inny niż: litera, cyfra, podkreślnik

12. Co oznacza \s w wyrażeniach regularnych?

Dowolny biały znak (spacja, ...)

13. Co oznacza \S w wyrażeniach regularnych?

Dowolny znak różny niż znak biały

14. Co oznacza \d w wyrażeniach regularnych?

Dowolna cyfra

15. Co oznacza \D w wyrażeniach regularnych?

Dowolny znak inny niż cyfra

16. Co oznacza **^** w wyrażeniach regularnych?

Dopasowanie musi się pojawić na początku tekstu

17. Co oznacza **\$** w wyrażeniach regularnych?

Dopasowanie musi się pojawić na końcu tekstu

18. Co oznacza **\b** w wyrażeniach regularnych?

Dopasowanie musi się pojawić na początku lub na końcu słowa.

19. Co oznacza **\B** w wyrażeniach regularnych?

Dopasowanie nie może się pojawić na początku lub na końcu słowa.

20. Zapisz wyrażenie regularne, które sprawdzi poprawność formatu wprowadzonego adresu www.

`^www\.[-\w]+(\.[-\w]+)+$`

21. Podaj wyrażenie regularne, które sprawdza poprawność nazwy użytkownika, która może składać się z małych liter, cyfr, myślnika, kropki oraz podkreślenia i ma mieć pomiędzy 2 a 10 znaków.

`^[a-z0-9_.-]{2,10}$`

22. Jakie są dwie najpopularniejsze składnie wyrażeń regularnych?

- a) uniksowa
- b) perlowa

~~Docker

23. Co to jest Docker?

Docker to platforma służąca tworzeniu, dostarczaniu oraz uruchamianiu aplikacji w zvirtualizowanym środowisku kontenerów.

24. Polecenie: **docker build ścieżka_do_pliku_Dockerfile** (np `/home/user/folder`)

Buduje obraz z pliku Dockerfile.

25. Wyjaśnij czym jest kontener w Dockerze?

- Jest to izolowana platforma uruchomieniowa dla aplikacji,
- Zawiera wszystko (system plików, biblioteki, zależności), co jest potrzebne aplikacji,
- Bazuje na jednym (bazowy) lub większej ilości obrazów (warstwy).

26. Podaj polecenie pozwalające na usuwanie zatrzymanego kontenera w dockerze.

`docker rm id_kontenera`

27. Do czego służy plik dockerfile?

Dockerfile służy do tworzenia pojedynczych kontenerów zawierających aplikacje.

//aga: Dockerfile to plik konfiguracyjny zawierający instrukcje potrzebne do budowy obrazu.

28. Czym jest narzędzie docker compose?

Jest to narzędzie, które umożliwia przy użyciu pliku docker-compose.yml pozwala na tworzenie oraz uruchamianie aplikacji wielokontenerowych.

29. Podaj komendę potrzebną do uruchomienia pliku docker-compose.yml.

docker-compose up

30. Podaj i wyjaśnij podstawowe instrukcje używane w pliku dockerfile:

- **FROM** - instrukcja FROM inicjuje nowy etap budowy i ustawia obraz podstawowy dla kolejnych instrukcji. Plik Dockerfile musi rozpoczynać się od instrukcji FROM.
- **RUN** - instrukcja RUN wykona zadanie polecenie w nowej warstwie i zatwierdzi wynik w obrazie. Zatwierdzony wynik zostanie wykorzystany do następnego kroku w pliku dockerfile.
- **WORKDIR** - instrukcja WORKDIR ustawia katalog roboczy dla dowolnych instrukcji RUN, CMD itd., które następują po nim w pliku Dockerfile. Jeśli WORKDIR nie istnieje, zostanie utworzony, nawet jeśli nie zostanie użyty w żadnej kolejnej instrukcji Dockerfile.
- **CMD** - w pliku Docker może znajdować się tylko jedna instrukcja CMD. Jeśli napiszesz więcej niż jedną, wtedy tylko ostatnia CMD zacznie obowiązywać. Głównym celem CMD jest zapewnienie wartości domyślnych dla kontenera wykonującego.

31. Co to jest Docker Hub?

Docker Hub to publiczny rejestr zawierający dużą liczbę gotowych do użycia obrazów(kontenerów).

32. Napisz, krótki plik dockerfile pobierający plik cpp i kompilujący do uruchomienia. (należy usunąć komentarze po prawej stronie komend "<<--").

FROM ubuntu:latest << -- instalacja ubuntu

RUN apt-get update && \

apt-get install g++ -q -y << -- instalacja gcc

ADD https://raw.githubusercontent.com/mbalcer/hello-world/master/hello.cpp

/home/Kontener << -- pobranie surowego pliku z gita

WORKDIR /home/Kontener << -- ścieżka w której działają skrypty min gcc

RUN g++ -o hello hello.cpp << -- kompilacja do pliku wykonywalnego: hello

CMD /home/Kontener/hello << -- uruchomienie skompilowanego pliku

~~GIT

33. Czym jest GIT?

Jest to rozproszony system kontroli wersji.

//2 wersja// Git to narzędzie służące do wspólnej, jednoczesnej pracy kilku użytkowników nad jedną aplikacją, bez konieczności ciągłego dostępu do sieci.

34. Podaj podstawową zaletę Rozproszonego Systemu Kontroli Wersji względem lokalnego i scentralizowanego systemu

Każdy użytkownik klonuje na swój komputer całe repozytorium z głównego serwera, dzięki czemu w przypadku jego awarii, nie paraliżuje to pracy, a repozytorium można przywrócić na główny serwer z każdej innej maszyny.

35. Jakie są trzy stany, w jakich mogą znajdować się pliki w Git?

Trzy stany plików – stany, w których mogą znajdować się pliki: zatwierdzony, zmodyfikowany i śledzony.

36. Czym jest przechowalnia?

Przechowalnia – jedna z głównych sekcji projektu Git, zawierająca informacje o tym, czego dotyczyć będzie następna operacja commit. Przeznaczona dla plików śledzonych.

37. Jak można rozpocząć projekt w gicie?

- Polecenie `git init` - w istniejącym katalogu utworzony zostanie podkatalog `.git`,
- Polecenie `git clone` - rozpocznie kopią istniejącego już repozytorium.

38. Polecenie: `git init`

Tworzy repozytorium GIT-a.

39. W jakim celu tworzy się gałęzie w Gicie?

Gałęzie tworzy się w celu rozwijania funkcjonalności odizolowanych od siebie.

40. Polecenie: `git branch nazwa_gałęzi`

Tworzy nową gałąź.

41. Polecenie: `git checkout nazwa_gałęzi`

Przechodzi na gałąź `nazwa_gałęzi`.

42. Polecenie: `git checkout -b nazwa_gałęzi`

Tworzy nową gałąź i przechodzi na nią.

43. Merging to?

Jest to scalanie gałęzi (branch'y) w projekcie GIT-a.

44. W jaki sposób można doprowadzić do zakleszczenia w Gicie?

Do zakleszczenia może dojść, gdy dwóch lub więcej użytkowników chce w tym samym czasie wysłać na serwer tą samą zmodyfikowaną gałąź.

45. Jaka jest różnica między `git pull` a `git fetch`?

Polecenie `git pull` pobiera wszystkie ostatnie zmiany ze zdalnego repozytorium i od razu je stosuje (merguje) w naszym lokalnym repozytorium (zastosowanie zmian dotyczy tylko gałęzi, w której aktualnie się znajdujemy). `Git fetch` pobiera zmiany ze zdalnego repozytorium lecz nie stosuje ich co pozwala na zapoznanie się ze zmianami. `Git fetch` nie wmergowuje automatycznie zmian do lokalnych gałęzi.

46. Czy do pracy na zdalnym repozytorium jest potrzebne stałe połączenie z Internetem? Dlaczego?

Do pracy nad zdalnym repozytorium nie jest potrzebne stałe połączenie z Internetem, ponieważ w komputerze posiadamy ostatnią pobraną kopię repozytorium, a połączenie z siecią jest potrzebne tylko do jej przesłania na serwer oraz pobrania zaktualizowanej wersji.

47. Branching

Jest to sposób pracy polegający na tworzeniu różnych wersji/gałęzi projektu.

~~Pozostałe

48. Co to jest test jednostkowy? **OK**

Jest to metoda testowania tworzonego oprogramowania poprzez wykonywanie testów weryfikujących poprawność działania pojedynczych elementów (jednostek) programu.

49. Do czego służą testy jednostkowe?

Jego zadaniem jest weryfikacja (bez ingerencji programisty), czy testowany kod działa poprawnie.

50. Czym jest debugowanie?

Proces redukowania błędów w oprogramowaniu, zazwyczaj polega to na kontrolowanym wykonywaniu programu pod nadzorem debuggera. Umożliwia badanie wartości zmiennych i różnych innych aspektów programu w celu wykrywania błędów i sprawdzania poprawności działania.

51. Wymień trzy zalety stosowania debuggerów.

- Efektywne śledzenie wartości poszczególnych zmiennych,
- wykonywanie instrukcji krok po kroku,
- wstrzymywanie działania programu w określonych miejscach.

52. Czym jest dekompiletor?

To Program przekształcający język maszynowy lub kod bajtowy do postaci języka wyższego poziomu.

53. Do czego służy profiler?

Profiler służy do dynamicznego analizowania programu pod kątem wydajności oraz zasobożerności, dostarcza nam wielu informacji o stworzonej przez nas aplikacji, z jego pomocą możemy sprawdzić wydajność naszego programu w środowisku np. NetBeans.

54. Co to jest ciasteczko?

Cookies są to niewielkie informacje przesyłane przez serwis internetowy, który odwiedzamy. Zawarte są w nim informacje o przeglądarce, lokalizacji, sesji i inne.

55. Co to są ciasteczka stron trzecich?

Ciasteczka są to informacje zamieszczane przez skrypty innych serwisów internetowych. Ciasteczka stron trzecich mogą być używane w celach marketingowych, dla przykładu logując się do konta google, a następnie na konto udemy, cena zwykłego kursu może się różnić gdyż firmy przekazują sobie informacje o użytkownikach oraz ich dane osobowe.

56. Rozwiń skrót IDE.

Zintegrowane środowisko programistyczne (Integrated Development Environment)

57. Co umożliwia nam narzędzie Selenium IDE?

Selenium IDE umożliwia tworzenie automatycznych testów stron internetowych.

58. Co umożliwiają scenariusze tworzone w Selenium IDE?

Umożliwiają zautomatyzowane działania na stronach WWW takie jak: logowanie, edycja formularzy, wyszukiwanie stron.

59. Do czego służy selenium w pythonie ?

Selenium w pythonie służy do uruchamiania skryptów przechodzących przez strony internetowe. Skrypt można wygenerować za pomocą zwykłego selenium i wygenerować kod do pythona który następnie możemy uruchomić z konsoli poleceniem `python skrypt.py`.

Do działania selenium w pythonie konieczne jest uruchomienie drivera geckodriver konieczne w katalogu `/usr/bin`.

60. Co to jest CI (Continuous Integration)?

Jest to praktyka programistyczna, w której członkowie zespołu często scalają swoje wyniki pracy. Dzięki temu powstaje kilka zintegrowanych wersji kodu, które są sprawdzane przez automatyczny proces budowania (i testowania). Mamy do tego takie narzędzia jak właśnie Jenkins.

61. Do czego służy zakładka Elements w Narzędziach Deweloperskich przeglądarki?

Dzięki niej mamy dostęp do drzewa DOM oraz jego elementów i ich właściwości. Możemy także dowolnie modyfikować elementy strony takie jak: tło, wielkość czcionki, czy też same informacje zawarte na stronie.

62. Jaka jest rola kontroli wersji?

System kontroli wersji śledzi wszystkie zmiany dokonywane na pliku (lub plikach) i umożliwia przywołanie dowolnej wcześniejszej wersji.

63. Modele wytwarzania oprogramowania:

- model wodospadowy
- model prototypowy
- model przyrostowy
- model spiralny
- programowanie ekstremalne
- programowanie zwinne
- scrum

63. Czym jest Maven?

Maven to narzędzie automatyzujące budowę oprogramowania, dostępne w Javie.

65. Podaj program, dzięki któremu możemy tworzyć wirtualne maszyny?

Oracle VM VirtualBox

66. Do czego służy htop?

Program służy do monitorowania obciążenia procesora, zajętości pamięci i pliku wymiany w czasie rzeczywistym. Umożliwia badanie działania aplikacji wielowątkowych.

67. Co to jest wstrzykiwanie kodu SQL?

Nieautoryzowane wpisanie kodu sql na stronie lub w aplikacji, np. formularz logowania.

68. Co to jest Node.js?

Jest to środowisko uruchomieniowe służące do tworzenia aplikacji na bazie języka javascript. Umożliwia tworzenie aplikacji sterowanych zdarzeniami wykorzystujących asynchroniczny system wejścia-wyjścia.

69. Jakie zakładki możemy znaleźć w narzędziach dla programistów w przeglądarce Chrome?

- a) elementy (np. podgląd drzewa DOM)
- b) zasoby
- c) sieć
- d) scripts (W tym miejscu, możemy w najprostszym skrócie podejrzeć i zdebugować kod naszych skryptów.)
- e) timeline wszystkie zdarzenia przeglądarki związane z wyświetlaniem strony osadzone na osi czasu
- f) profiles (m.in. zużycie procesora)

- g) audits
- h) console (znajdziemy informacje o błędach w JavaScriptcie, CSS i innych nieoczekiwanych przez nas zdarzeniach)
- i) pagespeed (Widoczny na ostatniej zakładce PageSpeed, to specjalne rozszerzenie dla *Narzędzi programistów*, które pozwala na mierzenie prędkości wczytywania strony oraz wyświetlanie porad, które mogą przyspieszyć ładowanie się naszej strony)

70. Co to jest interpreter?

Interpreter – program komputerowy, który analizuje kod źródłowy programu, a przeanalizowane fragmenty wykonuje. Wykonanie powtórnie tego samego fragmentu wymaga powtórnej analizy.

71. Co to jest kompilator?

Kompilator - to program tłumaczący program w języku wysokiego poziomu, który tworzy programy wynikowe, uruchamialne dopiero po zakończeniu tłumaczenia (w odróżnieniu od interpretera).

72. Do czego służy AJAX?

Korzystając z AJAX możemy przeładowywać części strony. A że robimy to zazwyczaj asynchronicznie, użytkownik w tym czasie dalej może korzystać z naszej strony.

73. Do czego służy Toogle Device Toolbar?

za pomocą opcji Toogle Device Toolbar możemy zaobserwować, jak strona zachowuje się dla różnych typów urządzeń (telefon/tablet itp).