Generics

№ уроку: 3 **Курс:** TypeScript

Засоби навчання: Visual Studio, Visual Studio Code, NotePad++

Огляд, мета та призначення уроку

Мета уроку – ознайомлення студентів із новими методами **ES5** для роботи з масивами. Вивчення узагальнених типів даних та їх призначення. Познайомити учнів із поняттям **Simbol** та генераторами.

Вивчивши матеріал даного заняття, учень зможе:

- Застосовувати вивчені методи для роботи з масивами
- Розуміти різницю **for of** та **for in**
- Використовувати узагальнення під час роботи з функціями
- Використовувати узагальнення під час роботи з класами
- Використовувати обмеження у **Gineric**
- Створювати генератори

Зміст уроку

1. Методи **ECMAScript 5** для роботи з масивами
2. Цикл **for of**
3. **Generic** або узагальнені типи
4. Використання обмежень в узагальнених типах
5. **[Symbol.Iterator]**
6. Генератори

Резюме

- **Generic** (узагальнений тип чи шаблон) – спеціальний тип даних, який дозволяє створювати компоненти, не прив'язуючись до конкретного типу даних, а вказуючи його під час створення компонента.
- **Symbol** — унікальний та незмінний тип даних, який може бути використаний як ідентифікатор для властивостей об'єктів. Крім символів, що визначаються користувачем, існують заздалегідь визначені вбудовані символи. Останні потрібні для відображення внутрішньої поведінки мови.
- **Symbol.iterator** – метод, що повертає ітератор за замовчанням для об'єкта. Використовується конструкцією **for...of**.
- До сучасного **JavaScript** додано нову концепцію «ітерованих» (**iterable**) об'єктів.
- **Об'єкти, що ітеруються –** це об'єкти, вміст яких можна перебрати в циклі. У ітератора повинен бути метод **next(),** який при кожному виклику повертає об'єкт із властивостями: **value** – чергове значення, **done** і **false**, якщо є ще значення, і **true** – в кінці.
- **Генератори** – новий вид функцій у сучасному **JavaScript**. Вони відрізняються від звичайних тим, що можуть призупиняти своє виконання, повертати проміжний результат і далі відновлювати його пізніше, у довільний час. Для оголошення генератора використається нова синтаксична конструкція: **function*.** Її називають «функція-генератор» (**generator function**). При її запуску код такої функції не виконується. Натомість вона повертає спеціальний об'єкт, який якраз і називають «генератором». Основним методом генератора є **next().** Під час виклику він відновлює виконання коду до

CyberBionic systematics

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

найближчого ключового слова **yield**. Після досягнення **yield** виконання припиняється, а значення повертається у зовнішній код. Повторний виклик **generator.next()** відновить виконання та поверне результат наступного **yield**.

Закріплення матеріалу

- Що таке узагальнений тип даних?
- Що таке ітератор?
- Що таке ключове слово yield?
- Який метод за замовчуванням є у генераторі?
- Що функція повертає генератор?
- Що таке Simbol?
- Що повертає метод next()?

Додаткове завдання

Створіть екземпляр класу, в конструктор якого користувач буде передавати рядкові значення. Встановіть в класі метод для визначення функції генератора, яка на кожному значенні у властивостях класу встановлює **yield**. При виклику даної функції з класу перевірте всі значення уведені користувачем та зупиніть перебір – у випадку якщо користувач ввів числове значення. Помилку виведіть у консоль.

Самостійна діяльність учня

Завдання 1
Вивчити основні поняття, розглянуті на уроці

Завдання 2
Створіть словник власних визначень, використовуючи **Generic function.** В середині повинні бути визначення для трьох властивостей – ключ, значення, опис (різних типів даних). Для отримання чи запису використовуйте **get/set** реалізації доступу. Також для полів необхідно використовувати модифікатори доступу (на ваш розсуд). В підсумку повинний вийти словник термінів, отримуючи у вхідний параметр різні типи даних для реалізації.

Рекомендовані ресурси

https://www.typescriptlang.org/

https://www.typescriptlang.org/play/index.html

https://github.com/Microsoft/TypeScript

CyberBionic systematics

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com