**УЧЕБНА ПРОГРАМА ПО “Основи на Програмирането с ДжаваСкрипт” ЗА IX КЛАС**

**(ВЪВЕЖДАНЕ НА ДИСЦИПЛИНАТА ПО ПРОЕКТ “ИНОВАТИВНО УЧИЛИЩЕ”)**

**КРАТКО ПРЕДСТАВЯНЕ НА УЧЕБНАТА ПРОГРАМА**

Обучението по **Основи на Програмирането с ДжаваСкрипт** в гимназиален етап е насочено към овладяване на базисни знания, умения и отношения, свързани с учебния предмет, с изграждането на дигитални компетентности на ученика и с приложението им в областта на дигиталните технологии.

В този клас се придобиват систематизирани знания и умения за основите на програмирането. Формират се нови знания и умения за писане на код в кодов редактор. Акцентът в обучението в IX клас е върху използването на адаптивни учебни материали, целящи да формират знания и умения за използване на основополагащи понятия в програмирането на различните програмни езици.

Учебното съдържание е представено в следните основни теми: 

* Първи стъпки с JavaScript. Работа с конзола

# JavaScript - Условни конструкции

* JavaScript - Цикли

# JavaScript - Масиви

# JavaScript - Функции

В програмата са включени въвеждащи теми за запознаване с видовете променливи, видовете цикли в прoграмирането, работа с условни конструкции и използване и създаване функции, работа в кодов редактор и конзола. Основната цел на тези теми е да представят набора от софтуерни средства, които ще бъдат изучавани и използвани в гимназиалния етап на обучение по Програмиране. Учебното съдържание се доразвива в програмите за следващите класове.

**ОЧАКВАНИ РЕЗУЛТАТИ ОТ ОБУЧЕНИЕТО В КРАЯ НА КЛАСА**

В края на обучението в IX клас ученикът: 

* разпознава основните записвания в кода на програмен език JavaScript, и обяснява тяхното предназначение
* демонстрира отношение на отговорен потребител при работа в Интернет среда
* реагира на съобщенията, извеждани от използваното приложение, и коригира своя код съобразно забележките
* прилага съответстващата българска терминология при описание на дейности, свързани със средата за програмиране
* описва и спазва правилата за безопасна работа с компютърна система 
* изброява причините за използване нa функции в програмирането
* Търси и открива причините за проблемен код: безкраен цикъл, неработещи функции, съобщения в конзолата и други.
* използва основните команди за писане на работещ и постигащ предварително зададена цел код
* обяснява основните понятия и дейности, свързани с използваните програми 
* изброява и описва предназначението на понятията: променливи, условни конструкции в програмирането, индекси, масиви, функции, параметри, команди.
* използва средствата за комуникиране в системата за кодиране, като изпраща съобщения 

**УЧЕБНО СЪДЪРЖАНИЕ**

|  |  |  |
| --- | --- | --- |
| **Теми** | **Компетентности като очаквани резултати от обучението** | **Нови понятия** |
| **Тема 1: Първи стъпки с JavaScript. Работа с конзола.** | | |
| **1.1 Въведение в програмирането.** | * посочва примери от ежедневието, в които се използват софтуерни продукти * дава описание на понятията команда, тест-функция * изброява и спазва правила за безопасна работа и коректна експлоатация на компютърната система  * стартира, използва и приключва работа със средата за визуално програмиране, ориентира се със системата за изпращане на съобщения и комункации * осъществява диалог със системата и останалите й потребители, като използва елементите на потребителския интерфейс | Код, команда, кодов редактор, екран на кодовия редактор, команди, команден ред,  *Забележка:* Понятията се въвеждат в темата, но оперирането с тях и затвърдяването им продължават във всички теми, свързани с използване на софтуерни приложения. |
| **1.2 Конзола. Предназначение, основни елементи.** | * посочва основните предназначения на конзолата. * Може да извежда информация в конзолата, чрез използване на функцията console.log() | Конзола, стандартни функции в JS, извеждане на информация в конзола |
| **1.3 Типове данни в програмирането** | * Може да обясни разликата между типовете данни:string, boolean, double, int. * Може да конкатенира променливи от различен тип: | Конкатениране, Променливи и типове данни, |
| **1.4 Променливи и коментари** | * Moже да дефинира променливи в JS * Може да използва коментари | Дефиниране на променлива, коментар |
| **1.5 Свойства и функции в JS** | * Знае предназначението и може да използва свойството .length * Използва правилно Функциите '.toLowerCase()' & '.toUpperCase() в JS код | Свойство в ДжаваСкрипт, функции |
| Тема 2: JavaScript - Условни конструкции | | |
| **2.1 Основни логически оператори** | * Може да изброи основните логически оператори * Разбира разликата в употребата на основните логически оператори: ||, !, != , * Трансформира пълен в съкратен запис и обратно, като използва оператора ?: | Логически оператори, условна конструкция |
| **2.2 Условна конструкция switch** | * Работи с условна конструкция switch * Може да обясни разликата между стандартна и вложена if конструкция | switch, вложена условна конструкция |
| **2.3 Метод prompt** | * Умее да използва метод prompt за въвеждане на данни в конзолата | вход на данни в конзола |
| **Тема 3: JavaScript - Цикли** | | |
| **3.1 Въведение в циклите в програмирането.** | * Може да обясни необходимостта от използването на цикли в програмирането * Запознат е със стандартната конструкция на циклите: условие на цикъл, тяло, прекратяване на работата на цикъл * Изброява основните видове цикли в JS * Разбира и посочва разликата в употребата на видовете цикли | Цикли, повторяемост на код, Итерация, завъртане на цикъл, видове цикли, |
| **3.2 Цикли while, do-while** | * Може да използва съвместно цикъл с метод prompt * Използва цикъл while съвместно с логическите оператори && и || * Използва и обяснява начина на работа на цикъл do-while | Логически оператори && и || |
| **3.3 Команди за прекъсване на цикъл. Безкраен цикъл** | * Използва командата break в своя код * Използва цикъл do-while в своя код * Разбира нуждата от Оптимизира чрез съкращаване на излишен код * Преценява кога да използва цикли и кога да използва обикновени команди, за постигане на по-кратък код | Безкраен цикъл, оптимизиране на код |
| **Тема 4: JavaScript - Масиви** | | |
| **4.1 Що е масив?** | * Разбира разликата между масив и отделен елемент на масива * Записва коректно индексите на отделните елементи на масива * Познава понятията масив, индекс, параметър * Разбира нуждата от използване на цикли за обхождане на масив * Разбира начина на функциониране на оператор за целочислено деление модуло * Умее да коригира грешки в код, състоящ се от комбинирани елементи: цикъл, масив, променливи | Масив, елементи на масив, индекс |
| **4.2 Декларация и инициализация на елементи** | * Изброява основните операции с масиви * Редактира предварително зададен код с използване на масив * Разбира необходимостта от използване на масиви | декларация на масив и по-късна инициализация, стойност на елемент |
| **4.3 Едномерен и двумерен масив в ДжаваСкрипт** | * Обяснява разликата между едномерен и двумерен масив * Посочва подходяща употреба на видовете масив * Дава пример за двумерен масив от практиката | едномерен масив, двумерен масив, |
| **4.4 Обхождане на масив** | * Обхожда двупосочно елементите на масив, чрез използване на цикъл for * Взема броя на елементите на масив, с цел обработка на елементите | обхождане на масив, булева стойност |
| **Тема 5: JavaScript - Функции** | | |
| **5.1 Въведение в използването на Функции в програмирането** | * Обяснява предназначението на използване на функция в конзолна среда * класифицира функциите според вида им като: функции без параметър, функции с един или повече параметри. * идентифицира ситуации, в които могат да се използват различни варианти на кода, за постигане на едно и също крайно решение. | Функция и метод, Извикване на функция |
| **5.2 Деклариране и извикване на функция** | * Умее да декларира функция с и без параметър * Умее да извиква функция в JS код на правилното място в кода * Познава начина на действие на функциите * Може да трансформира функция с параметър, във функция без параметър, и обратно. | функции с параметър, функции без параметър, деклариране на функция |
| **5.3 Функции с повече параметри** | * Работи с функции с повече параметри, като обяснява нуждата от допълнителни параметри. * Умее да оптимизира код, чрез използване на функция с параметър | Оптимизиране на код, функции с 3 параметъра |
| **5.4 Рекурсия. Обхват на променлива** | * Може да обясни понятието рекурсия като дава пример * Може да обясни понятието ‘обхват на променлива” като дава пример | рекурсия, обхват на променлива |

**ПРЕПОРЪЧИТЕЛНО ПРОЦЕНТНО РАЗПРЕДЕЛЕНИЕ НА ЗАДЪЛЖИТЕЛНИТЕ УЧЕБНИ ЧАСОВЕ ЗА ГОДИНАТА**

**Допълнителни уточнения за конкретния учебен предмет**

Обучението се осъществява в компютърна зала, като за всеки ученик има самостоятелно работно място.

Над 50% от часовете се организират под формата на комбиниран урок, по време на който учениците изпълняват практически задачи.

**Препоръчително разпределение на часовете:**

|  |  |
| --- | --- |
| **За нови знания и умения** | **30** |
| **За упражнения в лабораторна среда** | **56** |
| **За обобщение** | **6** |
| **За контролни работи** | **8** |

**СПЕЦИФИЧНИ МЕТОДИ И ФОРМИ ЗА ОЦЕНЯВАНЕ НА ПОСТИЖЕНИЯТА НА УЧЕНИЦИТЕ**

Проверката и оценката на знанията и уменията в обучението по информационни технологии трябва да бъдат насочени към измерване постигането на заложените в учебната програма очаквани резултати.

Очакваните резултати от обучението са свързани с усвояването на специфична за предмета терминология, практически умения за съставяне на код в кодов редактор, умения за аргументиране при избора на технологично средство.

Поради спецификата и разнообразния характер на очакваните резултати при оценяването на знанията и уменията на учениците могат да се използват различни методи и средства за проверка и оценка: 

* Тестове, съдържащи въпроси и задачи със структуриран отговор или с ограничена свобода на отговора. Подборът на тестовите задачи трябва да се съобрази с формулираните в учебната програма очаквани резултати. Тестовете дават възможност да се обхване по-голям обем от учебното съдържание за по-кратко време. Могат да се използват за установяване на входно и изходно равнище или контролно, проведено в рамките на 20-25 минути. 
* Решаване на практически задачи, решението на които се реализира на компютър в час. Този тип задачи може да съдържа отделни компоненти, които измерват усвояването на конкретни умения за работа с изучавания софтуер, умения за извличане на информация, умения за създаване на модели, умения за творческо трансформиране и представяне на различни видове информация в дигитален формат и др. 
* Изпълнение на практически задачи и малки проекти за домашна работа. 
* Представяне на кратко проучване на допълнителни източници по дадена тема от учебното съдържание. 
* Оценяване уменията при работа по проект въз основа на зададената роля на отделния ученик при изпълнение на проекта. 
* Портфолио, което може да съдържа решаваните от ученика практически задачи в часовете, домашни работи, проучвания по дадена тема, тестове. За оформянето на портфолиото учителят може да посочи кои от решаваните практически задачи ще бъдат задължително включени в него и да представи критерии за оценяване на отделните задачи и на портфолиото като цяло. Задачите, включени като задължителни компоненти, трябва да измерват постигането на формулираните в учебната програма очаквани резултати. Портфолиото може да включва и допълнителни задачи.

*Забележка:* Индивидуалното портфолио може да се използва за оценяване на отделен ученик, при условие че всеки ученик работи самостоятелно на компютър, или включва само компоненти, които ученикът разработва самостоятелно – домашни работи, проучвания, тестове.

**Съотношение при формиране на срочна и годишна оценка:**

|  |  |
| --- | --- |
| Текущи оценки от устни, от писмени и от практически изпитвания върху конкретна задача | 40% |
| Оценки от контролни (теоретични или практически) или изходно ниво | 30% |
| Оценки от работа по проекти и индивидуално портфолио по предварително зададени критерии, домашни работи | 30% |

**ДЕЙНОСТИ ЗА ПРИДОБИВАНЕ НА КЛЮЧОВИТЕ КОМПЕТЕНТНОСТИ, КАКТО И МЕЖДУПРЕДМЕТНИ ВРЪЗКИ**

***Дейности за цялата програма, които могат да се включват във всяка тема***

Дейности, свързани с развитие на умения за учене: 

* Поставяне на задачи за работа с фрагменти от учебните помагала или помощната информация с цел самостоятелно запознаване с елементи на изучавания материал и пограмен език. 
* Използване на демонстрации и експериментиране в адаптивната среда на изучаваното софтуерно приложение.

***Дейности, свързани с развитие на уменията за общуване на чужд език: ***

* Използване на английско-български и българо-английски речник за елементи от интерфейса на изучаваните софтуерни приложения.

***Примерни дейности за отделни раздели и теми***

|  |  |
| --- | --- |
| Ключови компетентности | Примерни дейности и междупредметни връзки |
| Компетентности в областта на българския език | * Въвеждане на текст в определените от дигиталната или визуалната среда места.    Общуване в писмен вид с останалите участници в системата за визуално програмиране.   * Анализиране на потенциалните възможности за решаването на конкретен проблем в адаптивната среда за програмиране * Коментиране на възможностите за решаване по различни начини на конкретна проблемна ситуация. * Създаване и записване на собствен текст - коментари - за поясняване на създадения код. |
| Умения за общуване на чужди езици | * Използване на команди и код, означения както на български, така и на английски език.  * Въвеждане на английските, съвместно с българските наименования, на основните елементи на изучавания приложен софтуер и интерфейс * Използване на последователност от латински букви и/или знаци за означаване на наименования на функции. |
| Математическа компетентност и основни компетентности в областта на природните науки и на технологиите | * Използване на математически оператори за съставяне на условни конструкции, масиви, функции, цикли и означаване на индекси; * Изчертаване на познати фигури чрез писане на подходящ код; * Използване на знаци за сравнение при съставяне на тялото на условна конструкция. |
| Дигитална компетентност | * Обработване на информация.  * Разглеждане на допълнителна информация в Интернет, свързана с възможностите на програмния език - видео материали, печатни, аудио материали. * Използване дигитална идентичност. * Прилагане правила за безопасна работа в дигитална среда, и защита на личния профил в Интернет пространството.  * Създаване на дигитално съдържание.  * Решаване на проблеми с използване на дигитални технологии * Изучаване на логиката на Основите на програмирането,чиито правила са в сила за всички програмни езици. |
| Умения за учене | * Търсене и обработване на информация от различни източници. * Откриване на грешки в собствен и чужд код * Предлагане на повече от едно вярно решение |
| Социални и граждански компетентности | * Съставяне на програмен код, съвместно със съученици, за постигане на обща цел (ситуации, които включват толерантно общуване и толериране на други виждания и решения). * Разглеждане на информация от сайтове, свързани с безопасно използване на интернет. * Разглеждане на информация от сайтове, свързани с възможностите за приложение на програмния език JavaScript. |
| Инициативност и предприемчивост | * Планиране, изготвяне и представяне на мултимедиен проект по зададена тема. Например: история на възникване на програмния език, възможни приложения на езика за създаване на софтуерни решения. Кои приложения, създадени с програмния език, аз лично използвам, и др. |
| Културна компетентност и умения за изразяване чрез творчество | * Предлагане на сюжет за образователна игра, чието решение включва изучаваните IT технологии. |
| Умения за подкрепа на устойчивото развитие и за здравословен начин на живот и спорт | * Изработване на проекти - създаване на персонално портфолио, което включва изучените до момента понятия и технологии в програмирането. * Предлагане на идеи за образователни игри с елементи от познати спортове и демонстриращи здравословен начин на живот и хранене. |