

Программа «Современное WEB- программирование»

Входной ассемент

Лекция 1.

Модуль 1. «Основы цифровых технологий»

и

Модуль 2. «Основы web-разработки»

"Основы цифровых технологий. Введение в профессию"

Данная дисциплина включает в себя следующие темы:

1. Введение в профессию:

Какие виды работ есть в ИТ сфере. Что такое «ИТ решение». Жизненный цикл (ЖЦ) «ИТ Решения» - водопадная и спиральная модель. Роли в процессе разработки ИТ Решения. ИТ роли в ИТ организации. ИТ Специальности – взгляд со стороны государства и со стороны работодателя. Уровень квалификации – взгляд со стороны государства и работодателя. Что нужно изучить, чтобы стать ИТ специалистом. Рекомендованный алгоритм входа в ИТ работу. Резюме и собеседование при приеме на ИТ работу. Лайфхаки при входе в ИТ.

2. Основы цифровых технологий:

2.1. Он-лайн сервисы в профессиональной среде:

Правила использования личных и корпоративных мессенджеров, личной и корпоративной почты. Основы протокола HTTP. Службы DNS: доменные имена, покупка доменных имен и настройка DNS записей. Инструментальные средства поддержки проектной деятельности: Trello, Miro, Integromat (Make). Lean canv.

2.2. Основные правила ведения и оформления документации в профессиональной среде.

Изучение основных требований к оформлению, указанных в ГОСТ 7.32. Знакомство с интерфейсом сервисов Google: Google Документы, Google Таблицы, Google Презентации. Рассмотрение аналога сервисов Google от Яндекс - Яндекс.Документы. Демонстрация использования сервисов на примере задач тестировщика. Создание базовой структуры презентации для защиты выпускной квалификационной работы по дополнительной программе обучения.

Введение в ИТ-профессию и цифровые технологии

Что разберём за 5 минут:

- ИТ-решения и их жизненный цикл.
- Ключевые роли в ИТ.
- Как войти в ИТ-профессию.
- Базовые онлайн-инструменты.
- Оформление документации.

ИТ-решение и его жизненный цикл

ИТ-решение — программно-аппаратный комплекс для решения бизнес-задачи (CRM, мобильное приложение, система мониторинга).

Модели жизненного цикла:

Водопадная (Waterfall)

Последовательность: анализ → проектирование → разработка → тестирование → внедрение → поддержка.

Плюс: чёткая документация. *Минус:* негибкость.

Сpirальная (Agile/Scrum)

Итерации по 1–4 недели с полным циклом в каждом спринте.

Плюс: быстрая адаптация. *Минус:* требует самоорганизации.

Вывод: выбор модели зависит от проекта и команды.

Роли в ИТ и вход в профессию

Ключевые роли в разработке:

- Бизнес-аналитик (требования).
- Архитектор (проектирование).
- Разработчик (frontend, backend, mobile – код).
- Тестировщик (QA – качество).
- DevOps (развертывание).
- Project Manager (управление).

Уровни квалификации:

Junior (0-1.5 года) → Middle (1.5-3 года) → Senior (3+ лет)

Как войти в ИТ:

- Выберите направление (разработка, тестирование, аналитика).
- Изучите основы (языки, инструменты, английский).
- Сделайте пет-проект (сайт, бот, тест-кейсы).
- Соберите портфолио (GitHub, личный сайт).
- Ищите стажировки / junior-позиции.

Совет: сочетайте обучение с практикой и нетворкингом.

Онлайн-сервисы в профсреде

Правила коммуникации:

- Корпоративная почта — для официальных вопросов.
- Мессенджеры (Teams, Slack) — для оперативных обсуждений.
- Личные аккаунты не используйте для работы.

Базовые инструменты:

- **Trello** — управление задачами (доски, карточки).
- **Miro** — визуализация (майндмэп, схемы).
- **Integromat (Make)** — автоматизация интеграций.
- **Lean Canvas** — шаблон для бизнес-модели.

DNS и домены:

- Доменное имя (example.com) покупают у регистратора.
- DNS-записи (A, CNAME, MX) связывают домен с IP и сервисами.

Оформление документации

ГОСТ 7.32 — основные требования:

- Структура: титульный лист, оглавление, введение, основная часть, заключение, список литературы, приложения.
- Шрифт: Times New Roman, 14 pt, интервал 1.5.
- Нумерация разделов, таблиц, рисунков.

Сервисы для работы:

- **Google Workspace** (Документы, Таблицы, Презентации): совместная работа, история изменений, экспорт в PDF/DOCX.
- **Яндекс Документы** — аналог с интеграцией в Яндекс Почту.

Практика: создайте презентацию ВКР в Google Презентациях по ГОСТ 7.32:

- 10–12 слайдов;
- логичная структура (цель, задачи, результаты, выводы);
- читаемые графики и подписи.

Введение в веб-разработку на HTML и CSS

Что разберём:

- Настройка среды разработки (VS Code).
- Основы HTML: структура и теги.
- Введение в CSS: способы подключения и базовые стили.
- Блочная модель CSS.
- Flexbox и Grid: основы вёрстки.
- Bootstrap: ускоренная вёрстка.
- Анимации: первые шаги.

Основы HTML и инструменты разработки

Visual Studio Code:

- Установка и настройка редактора
- Полезные расширения (Emmet, Live Server)
- Создание и структура HTML-документов

Базовые теги HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>Мой сайт</title>
</head>
<body>
  <h1>Заголовок</h1>
  <p>Параграф текста</p>
</body>
</html>
```

- Заголовки, параграфы, списки, ссылки
- Инструменты разработчика в браузере

Введение в CSS и способы подключения

Что такое CSS:

- Каскадные таблицы стилей
- Визуальное оформление HTML-элементов

Способы подключения:

1. Встроенные стили - через атрибут style
2. Внутренние - в теге <style>
3. Внешние - через <link href="styles.css">

Базовые свойства:

```
selector {  
    color: blue;  
    background: white;  
    font-size: 16px;  
    margin: 10px;  
}
```

Блочная модель и позиционирование

Блочная модель CSS:

- Content → Padding → Border → Margin
- Box-sizing: content-box vs border-box

Типы позиционирования:

- **Static** - обычный поток документа
- **Relative** - относительно обычной позиции
- **Absolute** - относительно ближайшего positioned предка
- **Fixed** - относительно окна браузера
- **Sticky** - гибрид relative и fixed

Flexbox для адаптивных макетов

Основы Flexbox:

- Одномерное выравнивание элементов
- Гибкое распределение пространства

Ключевые свойства:

```
.container {  
    display: flex;  
    justify-content: center; /* горизонталь */  
    align-items: center;    /* вертикаль */  
    flex-wrap: wrap;  
}
```

```
.item {  
    flex: 1; /* гибкий размер */  
}
```

- **justify-content** - выравнивание по главной оси
- **align-items** - выравнивание по поперечной оси

CSS Grid и Bootstrap

CSS Grid:

Двухмерная сетка для сложных макетов.

Ключевые свойства:

- `display: grid;`
- `grid-template-columns/grid-template-rows` — размеры колонок/строк.
- `gap` — отступы между элементами.

Пример сетки:

```
.grid {  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    gap: 20px;  
}
```

Bootstrap:

Фреймворк для быстрой вёрстки:

- Готовые классы (`container`, `row`, `col-6`).
- Адаптивность «из коробки».
- Интеграция Flexbox/Grid.

Старт с Bootstrap: подключите CSS через CDN в `<head>`.

Псевдоклассы и Bootstrap

Псевдоклассы и псевдоэлементы:

- **:hover** - при наведении
- **:focus** - при фокусе
- **:nth-child()** - выбор по порядку
- **::before, ::after** - добавление контента

Bootstrap Framework:

- Готовая система сеток
- Адаптивные классы (col-md-6, container)
- Встроенные компоненты и утилиты
- Использование Flexbox и Grid "под капотом"

Анимации и переходы

Анимации в CSS:

Два подхода:

1.Переходы (transition) — плавное изменение свойств.

```
button {  
    transition: background-color 0.3s;  
}  
button:hover {  
    background-color: red;  
}
```

2.Анимации (animation) — сложные эффекты по ключевым кадрам.

```
@keyframes fade {  
    0% { opacity: 0; }  
    100% { opacity: 1; }  
}  
.element {  
    animation: fade 2s;  
}
```

Практическое применение:

- Плавные переходы состояний
- Анимация появления элементов
- Улучшение пользовательского опыта

Далее: практикуйтесь! Создайте мини-сайт с использованием изученных технологий.

Основы графического и веб-дизайна

Что разберём:

- Знакомство с Figma.
- Адаптивное проектирование.
- Компоненты и дизайн-системы.
- Основы стиля: сетка, вёрстка, типографика.
- Работа с референсами.
- Практические инструменты.
- Итоги и дальнейшие шаги.

Знакомство с Figma

Почему Figma?

- Облачный редактор (работа в браузере).
- Совместная работа в реальном времени.
- Бесплатная базовая версия.
- Интеграция с разработчиками (код-генерация).

Рабочее пространство:

- *Canvas* — область дизайна.
- *Layers* — панель слоёв.
- *Properties* — настройки объектов.
- *Assets* — компоненты и стили.

Базовые инструменты:

- *Frame* — контейнеры для макетов.
- *Shape* — фигуры (прямоугольники, круги).
- *Text* — работа с текстом.
- *Pen* — создание кривых.
- *Prototype* — связь экранов.

Figma Community:

Библиотека шаблонов, плагинов и UI-китов (figma.com/community).

Основы работы с макетами и проектирование

Создание структуры макета:

- Настройка фреймов под разные устройства
- Работа с направляющими и сетками
- Группировка элементов и организация слоев

Autolayout - основа адаптивного дизайна:

[Автоматическое выравнивание]

- Вертикальное и горизонтальное распределение
 - Автоматические отступы и промежутки
 - Ресайзинг элементов при изменении контента
-
- Направление (Direction) - горизонтальное/вертикальное
 - Выравнивание (Alignment) и распределение (Distribution)
 - Отступы (Padding) и промежутки (Gap)

Инструменты адаптивного проектирования

Autolayout — основа адаптивности:

- Автоматически выравнивает элементы при изменении размера.
- Работает как Flexbox в CSS.

Как настроить:

1. Выделите элементы → нажмите Shift + A.
2. Задайте направление (горизонталь/вертикаль).
3. Настройте отступы (*Spacing*).

Адаптивные элементы:

- Используйте *Constraints* (лево/право, верх/низ) для привязки.
- Применяйте *Resizing* (Hug content, Fill container).

Кликабельные прототипы:

1. Перейдите во вкладку *Prototype*.
2. Свяжите фреймы стрелками.
3. Настройте анимации переходов (*Instant*, *Dissolve*, *Move in*).

Пример: создайте кнопку с hover-эффектом через Prototype.

Компоненты и дизайн-системы

Компоненты — основа консистентности:

- Позволяют повторно использовать элементы (кнопки, карточки).
- Изменения в главном компоненте применяются ко всем экземплярам.

Работа с компонентами:

- Создание главного компонента (Main Component)
- Использование экземпляров (Instances)
- Переопределение свойств экземпляров

Как создать компонент:

1. Выделите объект → нажмите Ctrl + Alt + K.
2. Назовите компонент (например, Button/Primary).

Варианты компонентов:

- Группируйте вариации (активная/неактивная кнопка).
- Используйте *Variants* для переключения состояний.

Переменные (Variables):

- Хранят цвета, отступы, шрифты.
- Упрощают обновление стилей.

UI Kit и дизайн-система:

- Набор готовых компонентов и правил.
- Пример: Google Material Design, Apple Human Interface.

Основы стиля: сетка и вёрстка

Сетка — каркас дизайна:

- Помогает выровнять элементы.
- Типы: колоночная (12-колоночная), модульная.

Как включить сетку в Figma:

- Выделите фрейм → в *Layout Grid* выберите тип (Columns/Rows).
- Настройте количество колонок и отступы.

Вёрстка — логика размещения:

- Соблюдайте иерархию (заголовки, подзаголовки).
- Оставляйте «воздух» (отступы между блоками).
- Используйте выравнивание по базовой линии.

Типографика:

- Шрифты: сочетайте не более 2–3 гарнитур.
- Размеры: заголовок (24–36 px), основной текст (14–16 px).
- Межстрочный интервал: 1,5× от размера шрифта.

Типографика и работа с референсами

Основы типографики:

- Выбор и сочетание шрифтов
- Иерархия текста (H1-H6, Body, Caption)
- Межстрочный интервал (Leading) и кернинг
- Длина строки и читабельность

• **Референсы — источник идей:**

- Анализируйте успешные интерфейсы (соцсети, банки, маркетплейсы).
- Ищите вдохновение на профессиональных площадках.

Площадки для референсов:

- **Dribbble** (dribbble.com) — короткие анимации, детали.
- **Behance** (behance.net) — полные кейсы, портфолио.
- **Dprofile** (dprofile.ru) — российский аналог Behance.

Как работать с референсами:

- Сохраняйте скриншоты в папку «Референсы».
- Выделяйте удачные решения (цвета, анимации, компоновка).
- Адаптируйте идеи под свой проект.

Совет: создайте в Figma страницу с референсами для быстрого доступа.

Итоги и практические шаги

Вы узнали:

- как работать в Figma (интерфейс, инструменты, компоненты);
- основы адаптивного дизайна (Autolayout, прототипы);
- принципы сетки, вёрстки и типографики;
- где искать референсы и как их использовать.

Практические задания:

1. Создайте макет сайта (главная страница + карточка товара).
2. Добавьте Autolayout для адаптивности.
3. Сделайте кликабельный прототип (2–3 экрана).
4. Соберите UI Kit из 5 компонентов (кнопка, поле ввода, карточка).

Далее:

- Изучите продвинутые плагины Figma (Unsplash, Iconify).
- Попробуйте экспортить макеты для разработчиков.
- Участвуйте в дизайн-челленджах (например, на Dribbble).

Ресурсы для самообучения:

- Официальная документация Figma.
- Курсы на Udemy/Skillbox по UX/UI.