

# Päivä 1: Ohjelmisto- kehitys ja HTML-kieli

---



MIMMIT KOODAA -BOOTCAMP 2020  
JANI JÄRVINEN, INTERTECHNO TRAINING OY

# Ohjelman sisällöstä ja tavoitteista

---

- Code Bootcamp (CBC) on tiivis ohjelmoinnin koulutusohjelma, jonka tavoitteena on opettaa ohjelmoinnin perusteet nykyaikaisilla tekniikoilla
- Koulutusohjelman aikana tutustumme eri ohjelmointikieliin ja välineisiin web-ohjelmoinnin teemalla
- Saat parhaan hyödyn koulutusohjelmasta, kun osallistut aktiivisesti keskusteluihin, teet harjoituksia ja kysyt aina, kun jokin asia on epäselvää tai kaipaavat apua
- Koulutusohjelman aikana aloitamme henkilökohtaisen demon kehittämisen
  - Demon aihe ja tekniikat ovat jokaisen vapaasti päätettävistä, ja demon tarkoitus on tukea työllistymistä
- Tervetuloa mukaan, ja ole aktiivinen erityisesti nyt, kun koulutukset pidetään verkkokoulutuksina!

# Sisällön teemat

---

- Päivät 1-3: Teema web-teknologiat: HTML5, CSS, JavaScript
  - Tutustutaan viimeisimpiin teknologioihin sekä ohjelmistokehittämisen kokonaiskuvaan
  - Läpikäytäviä tekniikoita mm.: HTML5, CSS3, JavaScript, TypeScript
  - Kehitysohjelmistona käytetään Visual Studio Codea (toimii Windows, Linux, Mac)
  - Harjoituksia
  - Demon kehittäminen alkaa.
- Päivät 4-6: Teema: C# ja Visual Studio
  - Tutustutaan Microsoftin .NET -kehittämiseen C#-ohjelmointikielellä
  - Ohjelmointilogiikan harjoituksia
  - Web-taustajärjestelmien ("backend") kehittämistä ASP.NET Core -tekniikalla
  - Kehitysohjelmistoina käytetään Visual Studio Codea (toimii Windows, Linux, Mac) sekä Visual Studiota (vain Windows)
  - Harjoituksia
  - Demon kehittäminen jatkuu.
- Päivät 7-8: Teema: Pilvipalvelut + DevOps
  - Tutustutaan moderneihin tuotantoympäristöihin ja ketteriin menetelmiin
  - Tutustutaan DevOps-käsitteeseen ja automaatioon
  - Harjoituksia ja demoja selainympäristössä, mm. Azure ja AWS.

# Kurssipäivistä

---

- Kurssin ensimmäiset päivät käsittelevät web-ohjelmoinnin perusteita
  - Useimmilla osallistujilla on jo jonkin verran kokemusta ohjelmoinnista, joten aikaa kannattaa käyttää itselle mahdollisimman tehokkaasti: jos jokin käsitelty aihe on jo tuttua, voit tutustua aiheen tiimoilta kehittyneempiin ominaisuuksiin tai tehdä harjoituksia ja tehtäviä
  - Kurssipäivien aikana saat myös vinkkejä syvemmälle menevään opiskeluun, joten kannattaa seurata myös linkkivinkkejä
- Sukellamme kuitenkin varsin nopeasti syvemmälle ohjelmointiin, joten tahti kiihtyy alun jälkeen
- Kurssipäivät ovat virallisesti klo 9–16, ja jakautuvat suunnilleen näin:
  - Taukoja pidetään n. tunnin välein
  - Lounastauko on n. klo 12–13
  - Harjoitusten aikana tehdään itsenäistä työskentelyä, ja kysyä voit milloin vain

# Verkkokoulutuksista

---

- Käytämme verkkokoulutuksiin Microsoftin Teams-sovellusta, ellei toisin sovita
- Osallistua voi joko selaimella (Firefox, Chrome, uusin Edge) tai erillisellä Teams-sovelluksella
- Teamsin kautta näet ja kuulet kouluttajan, sekä näet kouluttajan tietokoneen ruudun
- Kun osallistut Teams-koulutukseen, pyri pitämään mikrofoni pois päältä ("mute") ellet halua kysyä tai kommentoida jotakin
- Teams-ohjelmiston chat-ominaisuutta voi myös hyvin käyttää kysymyksiin
- Parhaiten kuuntelu ja osallistuminen onnistuu sankamikrofoni-kuulokkeilla, mutta kannettavan koneen mikrofonia ja kameraa voi hyvin käyttää, jos oma ympäristö on hiljainen
- Muista myös Slack-kanava, jossa voit kysyä ja keskustella!

# Ohjelmoinnin yleiskuva

---

# Mitä ovat ohjelmistot?

---

- Yksinkertaisista komennoista koostuvia laajoja kokonaisuuksia, joita voidaan ajaa tietokoneella (tai tietokoneen kaltaisella laitteella)
- Ohjelmistot käsittelevät käyttäjän syötettä ja näyttävät sen pohjalta tuloksia
- Tyypillisesti vain digitaalisessa (sähköisessä) muodossa



# Esimerkkejä ohjelmistoista

---

- Tietokoneella

- Word, Excel, PowerPoint, Photoshop
- Web-selain

- Puhelimessa

- Pienoisohjelmat eli appsit (engl. apps), esim. Facebook, Snapchat, Reittiopas
- Karttasovellukset, aikataulut, viestintä, pelit, jne.

- Verkossa

- Google, Facebook, Twitter, verkkopankki, jne.

- Pienissä ja vähän suuremmissakin laitteissa

- Pesukone, mikroaaltouuni, televisio
- Henkilöauto, bussi, lentokone
- Paperikone, konttinosturi, kerrostalo



# Mitä on ohjelmistokehittäminen?

---

- Ohjelmistojen kehittäminen on sitä työtä, jolla ohjelmistoja valmistetaan
- Laaja kuva
  - Projektiin tarvitaan ainakin kehittäjiä, suunnittelijoita ja testaajia
  - Kaupallinen puoli: myynti, markkinointi, tuki, hallinto, jne.
- Tekninen puoli
  - Varsinainen kehittäminen tehdään tietokoneella käyttäen apuna tähän tarkoitettuja ohjelmistoja
  - Tuloksena ohjelmakoodia eli lyhyesti koodia
  - Koodi suoritetaan jossakin tietokonelaitteistossa tai vaikkapa pilvipalvelussa
- Rakkaalla työllä on monta nimeä
  - Sovelluskehitys, ohjelmistokehitys, koodaus, ...

# Keskeisiä käsitteitä

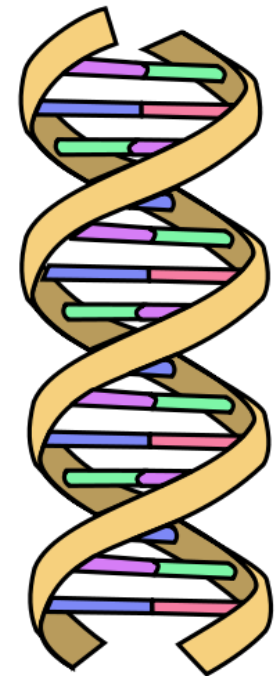
---

- Ohjelma eli tietokoneohjelma
  - Sarja ennalta laadittuja käskyjä, joita tietokone tai muu suorittimella varustettu laite osaa suorittaa
- Ohjelmointi
  - Se työ, jota tehdään kirjoitettaessa tietokoneohjelmia
- Sovellus tai ohjelmisto
  - Laajempi kokonaisuus, joka tyypillisesti ratkaisee jonkin laajemman ongelman tai tarjoaa käyttäjälleen tarvittavat välineet jonkin asian toteuttamiseksi
- Koodi
  - Ohjelma kirjoitettuna sellaiseen muotoon, että tietokone voi (tarvittaessa kääntäjän avustuksella) suorittaa ohjelman käskyt
- Suoritin eli prosessori
  - Fyysinen, elektroninen mikropiiri, joka suorittaa ohjelman koodin

# Laajempi kuva

---

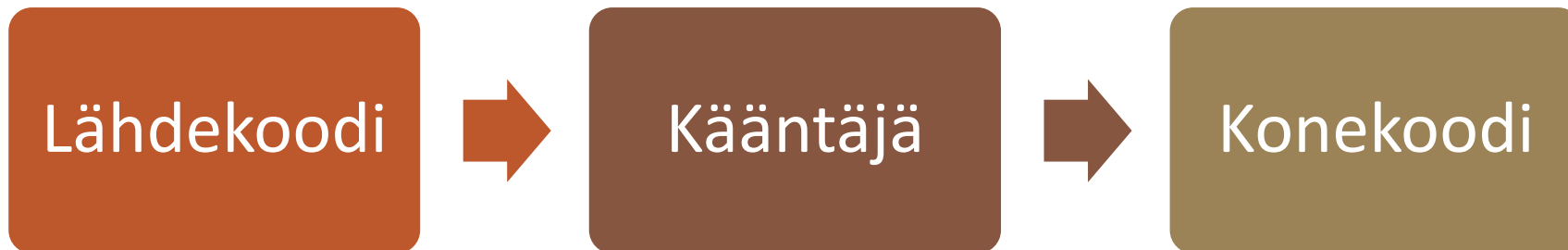
- Ohjelmia ja ohjelmointia on myös tietokonemaailman ulkopuolella



# Ohjelmointikielet

---

- Tietokoneiden ja muiden vastaavien laitteiden suorittimet ymmärtävät vain ykkösiä ja nollia
- Ihmisen on kuitenkin vaikeaa toteuttaa ohjelmia tällä tavalla, joten avuksi otetaan ohjelmointikieliä
  - Engl. *programming language*
- Kääntäjä kääntää lähdekooditiedoston ajettavaksi konekoodiksi, jota suoritin ymmärtää



# Erilaisia ohjelmointikieliä

---

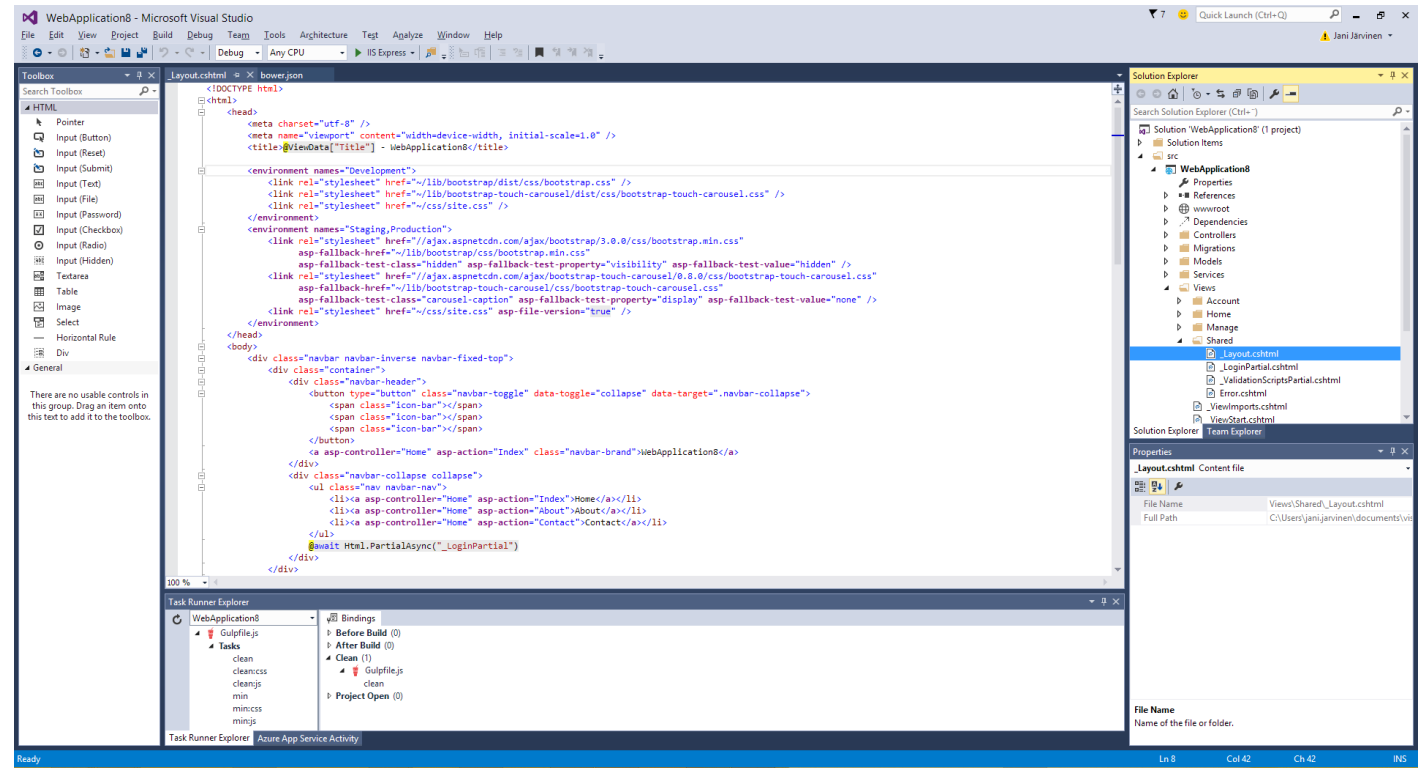
- Erilaisia ohjelmointikieliä ja niiden murteita on maailmassa tuhansittain
- Tyypillisesti jokainen ohjelmointikieli on suunniteltu tiettyä käyttötarkoitusta varten
- Ohjelmointikielellä kirjoitetaan lähdekoodia ja se on muodoltaan tyypillisesti pelkkää tekstiä
- Tunnettuja ohjelmointikieliä
  - C-perheen kieliä: C, C++, Java, C#, Objective-C
  - Skripti-pohjaisia: PHP, Python, JavaScript
  - Historiallisia: ALGOL, Fortran, COBOL, BASIC

```
namespace WebApplication8.Controllers
{
    [Authorize]
    2 references
    public class ManageController : Controller
    {
        private readonly UserManager<ApplicationUser> _userManager;
        private readonly SignInManager<ApplicationUser> _signInManager;
        private readonly IEmailSender _emailSender;
        private readonly ISmsSender _smsSender;

        0 references
        public ManageController(
            UserManager<ApplicationUser> userManager,
            SignInManager<ApplicationUser> signInManager,
            IEmailSender emailSender,
            ISmsSender smsSender)
        {
            _userManager = userManager;
            _signInManager = signInManager;
            _emailSender = emailSender;
            _smsSender = smsSender;
        }
    }
}
```

# Ohjelmistokehitin

- Ohjelmistokehitysväline eli lyhyesti kehitin on apuväline, jolla ohjelmointi helpottuu ja nopeutuu
- Tällä kurssilla käytetään Microsoftin valmistamaa Visual Studio -nimistä kehitysvälinettä
- Markkinoilla on tarjolla kymmenittäin erilaisia kehittämiä
- Termi kehitysympäristö
  - Engl. *development environment*



# Minimaalinen tietokone

---



# Ohjelmoijan työ

---

- Ohjelmoijan tai ohjelmistokehittäjän työ on usein vaativaa luovaa työtä
  - Mutta on muitakin kuin työ: henkinen ja fyysinen tasapaino, riittävä lepo, ruokavalio, liikunta, ...
- Vaatimuksia kehittäjille
  - Kyky hallita laajoja kokonaisuuksia, mutta silti kyetä keskittymään yksityiskohtiin
  - Kyky loogiseen päättelyyn ja ongelmien ratkaisuun
  - Kyky omaksua lähes jatkuvasti uutta
  - Tuntemusta siitä toimialasta, jolle ohjelmistoa ollaan kehittämässä
- Käytetty ohjelmointikieli ei juurikaan vaikuta siihen, millaisia taitoja tarvitaan
  - Kääntäen: kun on oppinut yhden ohjelmointikielen, on helppo oppia toinen
- Pienissä yrityksissä ohjelmoijat voivat harvoin keskittyä pelkästään ohjelmointiin, vaan mukaan tulee monia muitakin toimenkuvia
  - Esim. asiakaspalvelu, suunnittelu, tekninen tuki, IT-ylläpito, markkinointi, messut, myynti, ...



# Menestyvä kehittäjä

---

- Pelkkä ohjelmointitaito ei vielä riitä, mikäli haluaa menestyä IT-alan töissä
- Kuten monella muullakin alalla, seuraavista taidoista on aina apua
  - Aktiivinen ote tekemiseen: kysyy, ehdottaa, pyrkii jatkuvasti parantamaan, tekee muutakin kuin sen mitä pyydetään
  - Ymmärrystä muustakin kuin koodaamisesta: toimialaosaaminen, IT-osaaminen koodauksen ulkopuolella; esim. palvelimet, pilvipalvelut, automaatio, tietoturva, ...
  - Sovellusarkkitehtuuri: miten rakennetaan suurempia kokonaisuuksia tehokkaasti, turvallisesti ja taloudellisesti
  - Prosessiosaaminen: miten olla tiimin osana tai johtaa tiimiä, tehokkaat työtavat
- Tietojärjestelmiä kehitetään usein siksi, että rutiini- ja ihmistyötä voidaan automatisoida
  - Parhaimmillaan (tai pahimmillaan) hyvin toteutettu ratkaisu voi vähentää työpaikkoja, jopa omasi!

# Ei-teknisiä ominaisuuksia

---

- Menestyminen on muutakin kuin teknistä osaamista
- Sosiaaliset taidot: tulee toimeen muiden ihmisten kanssa, kyky käsitellä myös muiden mielipiteitä
- Viestinnälliset taidot: hyvä suullinen ja kirjallinen ilmaisu, Suomessa vähintään suomen- ja englanninkielen hyvä hallinta
- Ymmärrystä liiketoiminnasta: miksi yritys on olemassa, millainen on kustannusrakenne, ja mitä pyritään saamaan myydyksi, yrityksen strategia; yrittäjäyys
- Verkostot: monet työt ja palvelut ostetaan verkostojen, tuttujen ja aiempien kokemusten perusteella
- Riskinottokyky: jos pelaat aina varman päälle, et ole ensimmäisenä missään
- Kiinnostus ja halu oppia uutta: moni menestyvä kehittäjä käyttää paljon aikaa uuden oppimiseen

# HTML-kieli

---

# HTML-kielen perusteet

---

- Nimi on lyhenne sanoista HyperText Markup Language
- Koko modernin web-pohjaisen kehittämisen perusta: ei käyttöliittymää ilman HTML-koodia
- Tärkein ohjelmisto, joka osaa tulkita HTML-kieltä, on (web-)selain
- Tekstipohjainen kieli, jossa normaalin tekstin joukkoon lisätään koodauksia esimerkiksi linkeistä, otsikoista sekä alueista web-sivulla
- Alun perin HTML oli tarkoitettu tekstidokumenttien sisällön kuvaamiseen, mutta laajentunut huomattavasti sen jälkeen
- Läheisesti liittyviä kieliä tai standardeja
  - CSS: Cascading Style Sheets – ulkoasu, asemointi, fontit ja värit
  - JS: JavaScript – interaktiivinen toiminallisuus

# Esimerkki HTML-sivusta

---

```
<!DOCTYPE html>
<html>
<head>
  <title>Ensimmäinen sivu</title>
</head>
<body>
  <h1>Hei, maailma!</h1>
  <p>Tämä on tavallista tekstiä.</p>
</body>
</html>
```

# Harjoitus: Ensimmäinen HTML-sivu

---

- Toteutetaan ensimmäinen HTML-sivu, ja testataan sitä selaimessa
- Käytä työhön sopivaa tekstieditoria, joka tuottaa puhtaita tekstitiedostoja ("plain text")
- Esimerkiksi: Visual Studio Code, Atom, Notepad, UltraEdit, jne.

# Mistä lisätietoja?

---

- Muutamia hyödyllisiä verkkosivuja, joista voi oppia lisää HTML- ja web-tekniikoista
  - Perusteiden opetteluun
    - W3Schools: <https://www.w3schools.com/>
  - Pidemmälle eteenelle
    - Mozilla Developer Network eli MDN: <https://developer.mozilla.org/fi/>
  - Keskitason osaajalle
    - Tutorialspoint: <https://www.tutorialspoint.com/html5/index.htm>
- Taustajärjestelmien kehittämiseen käytämme kurssilla Microsoftin ASP.NET-nimistä tekniikka
  - Tämä ei ole vielä kuluva päivän aiheita, mutta voit halutessasi jo tutustua aiheeseen
  - <https://docs.microsoft.com/en-us/aspnet/core/>

# Ohjelmistokehitys toimialana

---



# Ohjelmistot ovat liiketoimintaa

---

- Harva yritys tai organisaatio voi nykypäivänä toimia ilman tietokoneita tai ohjelmistoja
- Nykyaikainen digitaalinen liiketoiminta ei olisi olemassa ilman ohjelmistoja
- Ohjelmistot voivat olla yritykselle hyvin tärkeä menestystekijä
- Suomessa ohjelmistoja kehittää ammatikseen arviolta 20 000 ihmistä
  - Esimerkiksi USA:ssa lähes 900 000
- Ohjelmistoja voi kehittää yksinkin, mutta usein mukaan liittyy työn eri osa-alueiden osaajia
  - Suurempien sovellusten kehittämiseen osallistuu helposti kymmeniä ihmisiä, toisinaan jopa satoja
- Pohdi, millaisia investointeja ohjelmistoliiketoiminnan aloittaminen vaatisi

# Esimerkki: pieni ohjelmisto

---

- Tavoite: web-pohjainen ohjelmisto, joka auttaa asiakasta omien laitteiden vianhallinnassa
- Hintaluokka: n. 20000 €
- Valmistumisaika: 3 kk
- Tiimi
  - Tekevällä yrityksellä: suunnittelija ja projektipäällikkö sekä kehittäjä
  - Asiakkaalla: kaupallinen yhteyshenkilö sekä IT-henkilö
- Lopputulos
  - Noin 10000 riviä ohjelmakoodia
  - 20 erilaista näkymää

# Toimenkuvia ohjelmistoyrityksissä

---

- Projektipäällikkö tai toimitusjohtaja
  - Neuvottelee asiakkaan kanssa tarvittavan ohjelmiston ominaisuuksista sekä hinnasta
  - On mukana ohjelmiston toiminnan suunnittelussa
- Arkkitehti
  - Suunnittelee ohjelmiston arkkitehtuurin, eli ratkaisun osat ja valitsee käyttöön sopivimmat välineet ja tekniikat
- Kehittäjät
  - Kirjoittavat ja tuottavat tarvittavan ohjelmakoodin, jotta ohjelma tekee sitä mitä asiakas toivoo sen tekevän
- Testaajat
  - Varmistavat, että ohjelma toimii kuten sen kuulukin, myös silloin kun esimerkiksi syöte on virheellistä
- Graafiset suunnittelijat
  - Toteuttavat ja suunnittelevat ohjelman ulkoasua ja käytettävyyttä

# Ohjelmistojen myynnistä

---

- Ohjelmistojen hinnat vaihtelevat ilmaisista maksullisiin, sentteistä miljooniin euroihin
- Ohjelmistojen myyntimalleja
  - Perinteinen käyttöoikeuden hankinta
  - Palveluna esim. kuukausimaksua vastaan
  - Ilmainen ohjelmisto, mutta mainosrahoitteinen
- Pohdi, miten voisit saada tuottamasi ohjelman myydyksi?
  - Suomessa?
  - Kansainvälisesti?

# Lopuksi

---

# Tehtäviä

---

1. Pohdi, onko ohjelmistolla fyysistä olomuotoa.
2. Etsi ainakin kolme esimerkkiä laitteista, joiden sisällä pyörii ohjelmisto. Jätä pois laskuista tietokoneet ja kännykät.
3. Selvitä, mitä välineitä ohjelmistojen kehittämiseen käytetään. Tarvitaanko kynää ja paperia lainkaan?
4. Mitä erilaisia työnkuvia ohjelmistojen kehittämiseen voi kuulua?
5. Etsi kaksi asiaa, jotka tulivat mahdollisiksi vasta ohjelmistojen myötä.

# Koodausta

---

1. HTML-koodi koostuu sisällöstä ja ohjauskomennoista, eli HTML-elementeistä tai ”tägeistä” (engl. tag) ja ne tunnistaa < ja > -merkeistä
  - Selvitä, mitä HTML-elementit ”a”, ”h1” ja ”link” tekevät?
2. Tee kaksi erillistä HTML-sivua tietokoneesi levyille samaan hakemistoon. Miten saat tehtyä hiirellä klikattavan linkin näiden kahden sivun välillä?
3. Lisää johonkin tekemääsi HTML-sivuun kuva. Missä kuvan tulee sijaita?
4. Lisää johonkin tekemääsi HTML-sivuun ”style”-niminen elementti. Mitä sen avulla voi tehdä?