ملخص الكورس التاسع

للبشمهندس الدكتور محمد أبو هدهود

من: عبدالرحمن محيى الدين

فهرس الدروس

اضغط علي الدرس للوصول اليه بسرعة

الشبكات والأجهزة

- 01 أساسيات الشبكات — LAN & WAN — Lesson 1
- 02 شبكة المدينة — MAN — Lesson 2
- 03 ما هو السيرفر؟ — Lesson 3
- 04 كيف تنتقل البيانات؟ — Lesson 4

البروتوكولات والعناوين

- 05 بروتوكول TCP — Lesson 5
- 06 IP Protocol — الجزء الأول (IPv4) — Lesson 6
- 07 IP Protocol — الجزء الثاني (IPv6) — Lesson 7
- 08 أنواع الـ IP — Lesson 8
- 09 Modem, Router, Gateway & Mesh — Lesson 9

10 DHCP — التوزيع التلقائي للـ IP — Lesson 10

11 NAT & IP Mapping — Lesson 11

12 ISP — مزود خدمة الإنترنت — Lesson 12

13 Ports & Socket — Lesson 13

14 Subnet Mask & CIDR — Lesson 14

15 MAC Address — Lesson 15

الإنترنت والويب

16 VPN — Lesson 16

17 Internet vs WWW — Lesson 17

18 Browsers — المتصفحات — Lesson 18

19 HTTP & HTTPS Protocols — Lesson 19

الدومين والعناوين

20 What is Domain Name? — Lesson 20

21 DNS — Domain Name Server — Lesson 21

22 Sub Domain — Lesson 22

23 URL — Uniform Resource Locator — Lesson 23

البروتوكولات والبيانات

24 FTP — File Transfer Protocol — Lesson 24

25 ?What is API — Lesson 25

26 ?What is XML — Lesson 26

27 ?What is JSON — Lesson 27

28 ?What is GUID — Lesson 28

29 Tier Architecture-3 — Lesson 29

لاحظ ان في بعض الدروس مشروحة باستخدام الذكاء الاصطناعي و البعض الاخر من تلخيصي لكلام الدكتور ولكن في الحالتين الشرح واضح و يذكر كل المعلومات التي ذكرها الاستاذ مع امثلة أوضح

Lesson 1

أساسيات الشبكات — الجزء الأول

ما هي الشبكة؟

الشبكة: مجموعة من الأجهزة المتصلة ببعضها البعض.

الهدف من الشبكة هو إن الأجهزة تقدر **تتواصل** مع بعضها و **تشارك الموارد** .

أمثلة على الاستخدام

- التحكم في التلفزيون عن طريق الموبايل.
- مشاركة الملفات بين أجهزة البيت.
- كل جهاز جديد بتدخله على الواي فاي بيبقى جزء من الشبكة.

أنواع الشبكات

الشبكات نوعين رئيسيين: **LAN** و **WAN**

LAN — الشبكة المحلية

LAN: Local Area Network — شبكة بتربط أجهزة قريبة من بعضها.

- ممكن تكون صغيرة زي شبكة البيت، أو كبيرة زي شبكة مستشفى أو مدرسة.
- المؤسسات الكبيرة ممكن يكون عندها أكتر من شبكة LAN — مثلاً شبكة لكل مبنى.

WAN — الشبكة الواسعة

WAN: Wide Area Network — شبكة بتربط شبكات LAN ببعضها على مسافات بعيدة.

- بتغطي مناطق جغرافية كبيرة — دول أو قارات.
- بتوصل آلاف أو ملايين الأجهزة.
- أشهر مثال عليها: الإنترنت نفسه.

العلاقة بين الاتنين: الـ WAN هي مجموعة شبكات LAN متصلة ببعضها على مسافات بعيدة.

كيف تتواصل الأجهزة مع بعضها؟

الاتصال بيكون بطريقتين:

النوع	الوصف	التقنية المستخدمة
سلكي	عن طريق كابلات باستخدام جهاز اسمه Switch	Ethernet
لاسلكي	عن طريق موجات الراديو ,باستخدام ال Router	(Wireless (Wi-Fi

البروتوكول

Protocol: مجموعة قواعد موحدة بتحدد إزاي الأجهزة تتواصل مع بعضها — سواء سلكي أو لاسلكي.

Ethernet

الـ Ethernet هو البروتوكول المستخدم في الاتصال السلكي — هو اللي بيحدد القواعد اللي الأجهزة بتتبعها لما تتكلم مع بعضها عن طريق كابلات.

Wi-Fi

الـ Wi-Fi هو أشهر طريقة للاتصال اللاسلكي بالشبكة.

Wi-Fi اختصار لـ Wireless Fidelity.

الفرق الأساسي: Ethernet أسرع وأكثر استقراراً، لكن Wi-Fi أسهل وبدون كابلات.

Lesson 2

الدرس الثاني: شبكة المدينة MAN

MAN: Metropolitan Area Network — شبكة حجمها متوسط، أكبر من LAN وأصغر من WAN، بتغطي مناطق زي المدن والبلدات.

مقارنة: MAN مقابل WAN

	MAN	WAN
السرعة	أسرع، تأخير أقل	أبطأ، تأخير أعلى
الأخطاء	ضوضاء وأخطاء أقل	ضوضاء وأخطاء أكتر
الازدحام	متوسط	عالي
التكلفة	تركيب أرخص	تركيب أغلى

العلاقة بين الثلاثة: LAN أصغرهم — داخل مبنى. MAN أكبر منه — داخل مدينة. WAN أكبرهم — بين دول وقارات.

لاحظ ان الMAN هي كمان بتتكون من شبكات LAN متصلة ببعضها ولكن عددها اقل من الWAN

Lesson 3

الدرس الثالث: ما هو السيرفر؟

Server: جهاز كمبيوتر بيستقبل ويخزن ويعالج ويشارك البيانات مع الأجهزة التانية في الشبكة.

ملاحظة مهمة

السيرفر مش لازم يكون جهاز هارد وير + سوفت وير مخصوص ,, ممكن يكون **برنامج** بيتشغل على جهازك العادي (سوفت وير فقط). مثلاً لو حطيت برنامج على اللابتوب بتاعك، اللابتوب نفسه بقى سيرفر.

مهمة السيرفر

جهازك لما يتحول ل خادم (سيرفر) ساعتها بيأدي مهمة السيرفرات في استقبال الطلبات او ال **Requests** و معالجتها و من ثَم ارسال الخدمة و او ما يسمى بالرود

مقارنة: Server مقابل PC

	Server	PC
الموثوقية	عالية جداً	أقل
الأداء	عالي	أقل
وقت التشغيل	يشتغل سنين بدون إيقاف	مش مصمم لكده
التكلفة	غالي	أرخص
صيانة	أقل	أكتر

أنواع السيرفرات

- ◈ **Web Server** — بيخدم صفحات الإنترنت
- ◈ **Database Server** — بيخزن وبيدير قواعد البيانات
- ◈ **Mail Server** — بيتولى إرسال واستقبال الإيميلات
- ◈ **File Server** — بيخزن الملفات ويشاركها على الشبكة
- ◈ **Application Server** — بيشغل التطبيقات
- ◈ **Game / VoIP / Image Server** — وغيرهم كتير

أشكال السيرفرات الفيزيائية

Tower: شكله زي الـ PC العادي — مستقل لوحده.

Rack: بيتركب في حامل معدني مخصوص.

Blade: بيتركب في Rack بس أرفع وأكثر كثافة.

الـ Rack بيخلي الشركات تحط عدد كبير من السيرفرات في مساحة صغيرة — ده اللي بتشوفه في data centers.

Lesson 4

How Data is Transferred?

فلنفترض ان في عدد من الكمبيوترات متصلة علي نفس الشبكة و في أحد الكمبيوترات أرسل ملف أو صورة لجهاز تاني , ايه اللي بيحصل بالترتيب

أولا كل الكلام اللي هنقوله ,مهما كان نوع الملف أو الشيء المرسل من جهاز لجهاز هيتم معاملته بنفس المعاملة : التحويل لكود بايني أولا و أبدا

الخطوات

- أولا الكمبيوتر مش بيكون عارف يمسك الصورة كجزء واحد فهو اولا بيقسم الصورة ل عدد من الاجزاء
- بيبعت جزء جزء للسرفر و الجزء الواحد بنسميه باكيت
- أول ما عدد الاجزاء توصل بيجمعهم مع بعضهم عشان يكونوا الصورة بشكل صحيح
-

بس ازاي بيعرف يجمعهم تاني صح؟

لانه كمان بيبعت معلوماتهم , ترتيبهم و طريقة وضعهم

مين اللي بيكون مسؤول عن كل العمليات دي؟

سيتم الاجابة في الدروس القادمة

الجزء اللي كان بيقص الصورة أو الملف لأجزاء معينة بالباينري و بيحولها من جهاز ل جهاز وكذلك علي الجهاز الاخر بياخد الاجزاء و يجمعها بنفس الطريقة اللي تم قصها هو :

بروتوكول TCP

TCP is responsible for :

- هو المسؤول عن ارسال الباكتات أو الحزم الصغيرة
- يعيد ترتيب البيانات المرسلة لاعادة تشكيلها بشكل صحيح
- ينظم الازدحام أو السرعة بين البيانات المرسلة من أجهزة لأخرى لتجنب حدوث انهيار في السستم

ولكن!

البروتوكول TCP يعمل علي الجهاز المرسل و الجهاز المستقبل لتسهيل نقل البيانات و لكن كيف يعرف الكمبيوتر عنوان الكمبيوتر الاخر لتحديد الجهاز الذي سيتم ارسال البيانات عليه؟

Lesson 6

IP Protocol Part 1

ازاي البيانات بتوصل لجهاز في موقعه المحدد؟

أي جهاز علي وجه الكرة الارضية متصل بأي شبكة أول ما بيشتغل بيكون عنده Address

و الAddress ده بنسميه **IP Address** بيمثل عنوان جهازك زي البصمة للانسان

ما هو الIP Address؟

- هو عنوان رقمى بيحدد عنوان أي جهاز في العالم زي ان كل بيت في العالم الحقيقي له عنوان
- يجعل الأجهزة تعرف من يُرسِل ومن يُرسَل إليه
- من أحد أشكاله **IPv4** و اللي بيتكون من أربع ارقام كل واحد فيهم بين 0 ل 255 مثلا **192.168.1.1**
- اختصار ل `Internet Protocol` و احنا عرفنا معنى البروتوكول قبل كدا

ما هو الIPv4؟

- الIP version 4 هو عبارة عن أدريس رقمي من أربع اجزاء كل جزء فيهم 8 بتات عشان كده كل رقم فيهم قيمته اخرها 255 عشان مساحة ال8 بتات اخرها 255
- ال8 بتات بتكون بايت واحد
- الأدريس كله بيكون 4 بايتس
- و هذا يضع عدد من الاحتمالات تساوي 4 مليار احتمال يعني يكفي 4 مليار جهاز

Lesson 7

IP Protocol Part 2 [IPv6]

عرفنا في اخر درس ان نوع العنوان IPv4 بيقدر يدي ادريس مميز ل 4.2 مليار جهاز أي : 2^{32} جهاز

ولكن هل هذا كافي لإحتواء جميع اجهزة العالم ؟ اكيد لا, هو كان كافي زمان بس

IPv6: هو الجيل الجديد من عناوين الانترنت , والذي تم اختراعه لان عناوين الانترنت القديمة IPv4 بدأت تنفذ بسبب كثرة الأجهزة الالكترونية

ما هو شكل الIPv6 ؟

- بيتم كتابته بالهكساديسمال بسبب كبره
- بيتكون من 8 أجزاء كل جزء 16 بت يعني بيكون في الاخر 128 بت أي : **16 بايت**
- بما ان الخانة الواحدة هكسا بتتكون من اربع خانات بايني يعني الجزء الواحد من الثمان اجزاء بيكون علي شكل أربع خانات هكسا
- مثال لشكل أحد الIPv6 **2001:0db8:0000:0000:ff00:0000:0042:8330**
- بيكون مسموحلك تشيل الاجزاء اللي بتكون قيمتها 0 فقط و تشيل الاصفار اليسارية و بيطلعلك النتيجة تكون نسخة أقصر **Sort version of IPv6**

2001:bd8:ff00:0:42:8330

- عدد الاجهزة الممكن تغطيتها بتكون $2^{128} = 3.4 \times 10^{38}$ جهاز

IP Part 3 (IP Types)

يوجد أربع أنواع من ال IPs :

1. **Public IP**

العنوان العام اللي بيشوفه العالم الخارجي.

بيتحدد من مزود الخدمة ISP وبيكون فريد على مستوى العالم.

ده اللي بيتعرف بيه جهازك على الإنترنت من برة.

2. **Private IP**

العنوان الداخلي جوه الشبكة المحلية زي الـ WiFi في البيت.

مش بيتشاف من برة الشبكة خالص.

ممكن يتكرر نفس العنوان في شبكات مختلفة.

3. **Static IP**

عنوان ثابت مش بيتغير خالص.

بيتستخدم للسيرفرات وكاميرات المراقبة والأجهزة اللي محتاجة عنوان ثابت دايماً.

4. **Dynamic IP**

عنوان بيتغير كل فترة أو كل ما تتوصل بالشبكة من جديد.

ده الأكثر شيوعاً عند المستخدمين العاديين.

مزود الخدمة بيحدده تلقائياً عن طريق بروتوكول الـ DHCP.

Lesson 9

What is Modem, Router, Gateway & Mesh Network?

ما هو الـ Modem؟

Modem: جهاز بيحول الإشارة القادمة من مزود الخدمة ISP عن طريق الكابل أو الـ DSL أو الفايبر لإشارة رقمية يقدر الكمبيوتر يفهمها.

الـ Modem لوحده قادر انه يتوصل بجهاز كمبيوتر واحد فقط عشان يوفرله الانترنت

الـ Modem بيكون موصول بالـ ISP من جهة، وبالـ Router أو الكمبيوتر مباشرة من الجهة التانية. وهو ده فعليا اللي بيوفر الانترنت ولكن مش هو اللي بيعمل شبكة, انما الراوتر اللي بيعمل شبكة تسمحلك تشبك عدد من الاجهزة

يعني ببساطة بيسمحلك تشبكه بجهاز واحد يا اما بكمبيوتر يا اما براوتر يوزعه علي البيت

ما هو الـ Router؟

Router: جهاز بيوزع الإنترنت على الأجهزة الموجودة في الشبكة المحلية، سواء عن طريق كابل أو لاسلكي.

الفرق بين Modem و Router

	Modem	Router
الوظيفة	يوصلك بالإنترنت	يوزع الإنترنت على الأجهزة
الاتصال	ISP ← جهازك	Modem ← أجهزتك
الشبكة	WAN	LAN

الراوتر بيديك أحد الشبكتين يا اما GHz 2.4 يا اما GHz 5 و ده بيكون الفرق بينهم :

	GHz 2.4	GHz 5
السرعة	أقل	أعلى
المدى	أبعد	أقصر
اختراق الحيطان	قوي	ضعيف
التشويش	عالي	قليل

	GHz 2.4	GHz 5
الأفضل لـ	البعد والتصفح	السرعة والجيمينج

ما هو الـ Gateway؟

Gateway: جهاز بيشبه الراوتر في الشكل و الاستخدام لكنه بيربط بين شبكتين مختلفتين في البروتوكول، وبيحول البيانات من صيغة لصيغة تانية عشان الشبكتين يقدروا يتكلموا مع بعض.

لاحظ ان الGateway ذات نفسه عبارة عن كمبيوتر و زي ما كل جهاز في العالم له IP Address هو كمان له IP Address

خصائص الـ Gateway

- ◈ بيتوصل علي طول بال ISP مش بيحتاج يتوصل بال Modem
- ◈ بيقدر يحول البيانات من بروتوكول لبروتوكول تاني
- ◈ الـ Router ممكن يأدي بعض وظائف الـ Gateway
- ◈ بيستخدم لربط الشبكة الداخلية بالإنترنت في الشركات الكبيرة

هل الـ Gateway نفس الـ Router؟

الـ Router بيبعت ويستقبل الباكيتات بين الشبكات، أما الـ Gateway فبيربط شبكات مختلفة البروتوكول تماماً — يعني الـ Gateway أشمل وأقوى في المهمة دي.

امتى تستخدم إيه؟

- ◈ لو الأجهزة على نفس الشبكة وليها بروتوكول واحد -> Router يكفي
- ◈ لو عايز تربط شبكتين ببروتوكولات مختلفة -> Gateway هو الحل

ما هو الـ Mesh Network؟

Mesh Network: نظام شبكة لاسلكية بيتكون من Router رئيسي وعدد من النقاط الصغيرة Nodes منتشرة في المكان، كل نقطة بتكمل تغطية النقطة اللي قبلها.

إزاي بيشتغل الـ Mesh؟

- ◈ في Router رئيسي موصول بالـ Modem زي العادي
- ◈ حواليه عدد من الـ **Nodes** أو النقاط الصغيرة منتشرة في البيت أو المكتب
- ◈ كل Node بتتكلم مع اللي جنبها وبتنقل الإشارة منها للتانية
- ◈ النتيجة: إشارة واي فاي قوية في كل ركن من غير أي بقع عمياء

الـ Mesh مفيد جداً في البيوت الكبيرة أو الأماكن اللي فيها حوائط كتير بتضعف الإشارة.

مقارنة: Router عادي مقابل Mesh

	Router عادي	Mesh Network
التغطية	محدودة	واسعة جداً
الإشارة	بتضعف مع البعد	ثابتة في كل مكان
التكلفة	أرخص	أغلى
الإعداد	سهل	أسهل من المتوقع
عدد الأجهزة	جهاز واحد	أكتر من جهاز

Lesson 10

IP Part 4 (DHCP)

ما هو الـ DHCP ؟

DHCP: اختصار لـ Dynamic Host Configuration Protocol — بروتوكول مسؤول عن توزيع عناوين الـ IP تلقائياً على الأجهزة داخل الشبكة.

بدل ما تروح لكل جهاز في البيت وتكتب له IP يدوي...
الـ DHCP بيعمل كل ده لوحده بشكل أوتوماتيك.

وظيفة الـ DHCP

- يدي كل جهاز داخل الشبكة IP Address تلقائي.
- يمنع تكرار نفس الـ IP بين جهازين.
- يوفر إعدادات الشبكة كاملة للجهاز.

يعني ببساطة: أول ما جهازك يتوصل بالواي فاي، الراوتر يبعتهله IP تلقائي عن طريق DHCP.

لاحظ ان الIP اللي بيتوزع بيكون Dynamic يعني مش هيفضل دايما معاك و غالبا بيكون لمدة اسبوع و ممكن علي حسب اعداداتك تطول المدة

مين اللي بيشغل DHCP ؟

غالباً اللي بيقوم بالمهمة دي هو:

- الـ Router في البيت
- أو Server مخصوص في الشركات الكبيرة

الراوتر هنا بيشتغل كـ **DHCP Server**.

البيانات اللي DHCP بيديها للجهاز

لما جهازك يتصل بالشبكة، مش بياخد IP بس، ده بياخد:

- IP Address
- Subnet Mask
- Default Gateway
- DNS Server

يعني كل إعدادات الشبكة اللي تخليه يطلع إنترنت.

مثال واقعي

لما تشغل الواي فاي في موبايلك:

1. الموبايل يطلب IP من الشبكة
2. الراوتر (DHCP Server) يرد عليه
3. يديه IP فاضي غير مستخدم
4. الموبايل يدخل الشبكة والإنترنت يشتغل

كل ده بيحصل في ثواني من غير ما تحس.

بدون DHCP

لو مفيش DHCP:

- ◈ هتكتب IP لكل جهاز يدوي
- ◈ ممكن يحصل تعارض IP
- ◈ الشبكة تبقى صعبة الإدارة

عشان كده DHCP أساسي في أي شبكة حديثة.

أي شبكة فيها أجهزة كتير مستحيل تشتغل عملياً من غير DHCP.

معظم الأجهزة في البيت بتاخد IP ديناميك عن طريق DHCP، أما السيرفرات والكاميرات غالباً Static.

Lesson 11

NAT & IP Mapping

ما هو الـ NAT ؟

NAT: تقنية داخل الراوتر تسمح لكل أجهزة الشبكة باستخدام Public IP واحد عند الاتصال بالإنترنت.

كل جهاز داخل الشبكة يمتلك Private IP خاص به

لكن عند الدخول للإنترنت يظهر بعنوان واحد فقط: Public IP الخاص بالراوتر.

بالتالي كل أجهزة الشبكة تظهر للإنترنت كأنها جهاز واحد فقط.

الفكرة الأساسية

الراوتر يعمل كوسيط بين:

◈ الشبكة الداخلية (Private IPs)

◈ الإنترنت (Public IP)

أي اتصال يخرج من أي جهاز داخل الشبكة يمر على الراوتر
والراوتر هو الذي يمثل كل الأجهزة أمام الإنترنت.

دور الـ Ports

التمييز بين الأجهزة والبرامج يتم باستخدام:

◈ IP Address

◈ Port Number

كل برنامج على الجهاز يعمل على Port مختلف،
والراوتر يستخدم الـ Port لمعرفة أي برنامج وأي جهاز أرسل الطلب.

لهذا يمكن لعشرات الأجهزة استخدام نفس Public IP بدون تعارض. (هنتكلم عن الموضوع ده ف الدرس 13)

ما هو IP Mapping ؟

IP Mapping: توجيه اتصال قادم من الإنترنت لجهاز محدد داخل الشبكة.

افتراضياً الإنترنت لا يستطيع الدخول مباشرة لأي جهاز داخلي
لأن الراوتر يخفي الأجهزة خلف Public IP واحد.

عند الحاجة للوصول لجهاز داخلي يتم عمل Mapping.

فكرته

يتم ربط:

◈ Port أو Public IP خارجي

بـ

◈ جهاز محدد داخل الشبكة (Private IP)

وأي طلب يصل لهذا الـ Port يتم تحويله مباشرة للجهاز.

يستخدم في

◈ كاميرات المراقبة

◈ Game Servers

- Remote Desktop
- استضافة موقع من المنزل

الفرق المختصر

	NAT	IP Mapping
الوظيفة	خروج الأجهزة للإنترنت	دخول الإنترنت لجهاز محدد
الاتجاه	من الداخل للخارج	من الخارج للداخل
الإعداد	تلقائي	يدوي
الاستخدام	موجود دائماً	عند الحاجة

NAT أساسي في أي شبكة، أما IP Mapping يستخدم فقط عند الحاجة للوصول لجهاز داخل الشبكة من الخارج.

Lesson 12

ISP

(Internet Service Provider)

الـISP (مزود خدمة الانترنت) هو ببساطة شديدة : الشركة اللي بتوصلك بالانترنت , زي وي وفودافون

وظيفة الـ ISP

- توصيل الإنترنت للبيت أو الشركة
- إعطاء الشبكة Public IP
- ربطك بالإنترنت العالمي

بدون ISP لا يمكن لأي جهاز الوصول للإنترنت حتى لو الشبكة الداخلية تعمل.

كيف يصل الإنترنت إليك؟

الاتصال يكون بهذا الترتيب:

جهازك --> الراوتر --> ISP --> الإنترنت العالمي

الـ ISP هو الجهة التي تنقل البيانات من شبكتك المحلية إلى بقية الإنترنت.

ماذا يعطيك الـ ISP ؟

عند الاشتراك في الإنترنت تحصل على:

- اتصال بالإنترنت
- Public IP للشبكة
- سرعة محددة حسب الباقة

الراوتر ينشئ شبكة داخل البيت، لكن الـ ISP هو الذي يربط هذه الشبكة بالإنترنت العالمي.

بدون ISP

- الأجهزة ترى بعضها داخل الشبكة فقط
- لا يوجد إنترنت
- لا يمكن فتح مواقع أو استخدام خدمات أونلاين

الـ ISP هو البوابة التي تخرج منها الشبكة المحلية إلى الإنترنت العالمي.

Lesson 13

Ports & Socket

ما هي الـ Ports ؟

Port: رقم منطقي داخل الجهاز يحدد البرنامج أو الخدمة التي ستستقبل البيانات.

نحن لا نتحدث هنا عن منافذ الهاردوير بل عن Software Ports داخل النظام.

كل برنامج أو سيرفر يعمل على Port معين ليستطيع إرسال واستقبال البيانات.

الفكرة ببساطة

الـ IP Address يوصل البيانات للجهاز الصحيح

لكن لا يحدد أي برنامج داخل هذا الجهاز سيستقبلها.

هنا يأتي دور الـ Port .

الـ IP يوصل للعمارة لكن الـ Port يحدد الشقة أو الدور الذي ستذهب إليه البيانات.

مثال

لو جهاز عليه:

- Web Server
- Email Server

ممكن يكونوا:

علي سيرفرين مختلفين أو علي نفس السيرفر

وفي الحالتين يتم التمييز بينهم عن طريق الـ Ports.

مثلاً:

- Web Server يعمل على Port 80
- Email Server يعمل على Port 25

عندما تصل البيانات للجهاز

الـ Port يحدد أي خدمة تستقبلها.

مثال توضيحي بعنوان العمارة

تخيل:

◈ IP Address = عنوان العمارة

◈ Port = رقم الشقة

ساعي البريد وصل للعمارة (IP)

لكن بدون رقم الشقة (Port)

لن يعرف أين يسلم الرسالة.

ما هو Socket ؟

Socket: هو الجمع بين IP Address و Port Number لتحديد اتصال كامل بين جهازين.

أي اتصال على الإنترنت يتم من خلال:

Socket Address = IP + Port

مثال رقمي

جهاز سيرفر عنوانه: **192.168.1.10**

والـ Web Server يعمل على: **Port 80**

إذن عنوان السوكيت يكون: **192.168.1.10:80**

هذا يعني: الاتصال سيذهب إلى جهاز محدد وإلى برنامج محدد داخل هذا الجهاز.

مثال اتصال كامل

عند فتح موقع:

◈ جهازك: `192.168.1.5:50000`

◈ السيرفر: `142.250.190.14:80`

كل طرف لديه Socket خاص به

ومن خلالهما يتم الاتصال بين البرنامجين.

أي اتصال شبكي لا يتم باستخدام IP فقط، بل باستخدام Socket (IP + Port).

Lesson 14

Subnet Mask & CIDR

فكرة تقسيم الـ IP Address

كل جهاز على الشبكة له IP Address ، لكن هذا العنوان ليس عشوائياً، بل مقسوم لجزئين أساسيين:

(1) Network Identifier → يحدد الشبكة

(2) Host Identifier → يحدد الجهاز داخل الشبكة

يعني الـ IP لا يحدد الجهاز فقط، بل يحدد أيضاً الشبكة التي ينتمي إليها.

مثال

192.168.1.10 هذا العنوان يحتوي:

◈ جزء يمثل الشبكة

◈ جزء يمثل الجهاز داخل الشبكة

لكن كيف يعرف الكمبيوتر أين ينتهي جزء الشبكة ويبدأ جزء الجهاز؟ هنا يظهر دور Subnet Mask .

ما هو Subnet Mask ؟

Subnet Mask: رقم يحدد أي جزء من الـ IP يمثل الشبكة وأي جزء يمثل الجهاز داخلها.

هو الذي يفصل بين: Network ID و Host ID وبدونه الجهاز لا يستطيع تحديد هل الجهاز الآخر داخل نفس الشبكة أم خارجها.

مثال واضح

IP: 192.168.1.10

Subnet Mask: 255.255.255.0

هذا يعني:

◈ أول 3 أرقام للشبكة

◈ آخر رقم للجهاز

إذن: Host → 10 Network → 192.168.1

أي جهاز يبدأ بـ 192.168.1 يكون داخل نفس الشبكة.

ما هو CIDR ؟

بدلاً من كتابة Subnet Mask كامل يستخدم شكل مختصر يسمى CIDR .

يكتب هكذا: 192.168.1.10/24

الرقم بعد / يمثل عدد البتات الخاصة بالشبكة.

/24 يعني: أول 24 بت للشبكة والباقي للأجهزة وهو نفس معنى: 255.255.255.0

تقسيمات الشبكات الشائعة

لتسهيل الفهم تم تقسيم الشبكات قديماً إلى Classes مشهورة:

Class A

/8 ← أول رقم للشبكة والباقي للأجهزة **شبكات ضخمة جداً**

مثال: 10.0.0.1

Class B

/16 ← أول رقمين للشبكة والباقي للأجهزة **شبكات متوسطة**

مثال: 172.16.5.1

Class C

/24 ← أول 3 أرقام للشبكة وآخر رقم للأجهزة **الأكثر استخداماً في المنازل والشركات**

مثال: 192.168.1.1

الخلاصة

الـ IP Address ينقسم إلى شبكة وجهاز

الـ Subnet Mask يحدد مكان التقسيم

الـ CIDR طريقة مختصرة لكتابة هذا التقسيم

> **Subnet Mask هو الذي يجعل الأجهزة تعرف هل هي داخل نفس الشبكة أم تحتاج المرور عبر الراوتر للوصول لشبكة أخرى.**

Lesson 15

What is MAC Address?

ما هو الـ MAC Address ؟

MAC Address: عنوان فريد لكل كارت شبكة في العالم يتم تعيينه من الشركة المصنعة.

أي جهاز يمكنه الاتصال بالشبكة يملك MAC Address خاص به:

- ◈ كمبيوتر
- ◈ موبايل
- ◈ راوتر
- ◈ طابعة
- ◈ أي جهاز متصل بالشبكة

لا يوجد جهازان في العالم لهما نفس MAC.

فكرته الأساسية

الـ IP Address يمكن أن يتغير لكن الـ MAC Address ثابت للجهاز.

يعني:

- ◈ IP = عنوان مؤقت داخل الشبكة
- ◈ MAC = هوية الجهاز الحقيقية

الـ MAC يشبه الرقم القومي للجهاز، بينما الـ IP يشبه عنوانه الحالي.

شكل الـ MAC Address

يكتب بالهيكساديسمال (أرقام + حروف):

00:1A:2B:3C:4D:5E

يتكون من 6 أجزاء كل جزء يمثل بايت (8 بت)

معلومة مهمة

أول 3 أجزاء تشير إلى الشركة المصنعة آخر 3 أجزاء رقم خاص بالجهاز نفسه.

لذلك يمكن معرفة الشركة المصنعة من الـ MAC.

متى يستخدم MAC Address ؟

يستخدم داخل الشبكة المحلية LAN فقط.

عند إرسال بيانات داخل نفس الشبكة:

◈ الجهاز لا يعتمد على IP فقط

◈ بل يستخدم MAC لتحديد الجهاز بدقة

أما عند الخروج للإنترنت:

◈ يتم الاعتماد على IP Address

الـ IP يحدد مكان الجهاز على الشبكة، أما MAC فيحدد هوية الجهاز نفسه داخل الشبكة المحلية.

Lesson 16

VPN

الفكرة الأساسية

VPN: اختصار لـ Virtual Private Network — شبكة افتراضية خاصة بتخليك تتصل بالإنترنت عن طريق سيرفر تاني بدل ما تتصل مباشرة.

تخيل إنك بدل ما تمشي في الشارع بوجهك وكل الناس تشوفك، بتلبس تنكر كامل ومفيش حد يعرف مين أنت أو جاي منين.

ده بالظبط اللي بيعمله الـ VPN .

إزاي بيشتغل؟

بدون VPN

◈ جهازك ← ISP ← الإنترنت

◈ الـ ISP شايف كل حاجة بتعملها

◈ المواقع بتشوف الـ IP الحقيقي بتاعك

مع VPN

◈ جهازك ← VPN Server ← الإنترنت

◈ الاتصال بيتشفّر قبل ما يطلع من جهازك

◈ المواقع بتشوف IP الـ VPN مش IP بتاعك

◈ الـ ISP بيشوف إنك متصل بس مش بيعرف بتعمل إيه

الـ VPN بيعمل نفق مشفر (Encrypted Tunnel) بين جهازك والسيرفر — أي بيانات بتعدي فيه مش حد يقدر يفهمها من برة.

خصائص الـ VPN

الخصوصية

مواقع الإنترنت مش بتشوف IP الحقيقي بتاعك، بتشوف IP الـ VPN Server بس.

التشفير

البيانات بتتشفر وأي حد يحاول يتنصت على الاتصال مش هيفهم أي حاجة.

تغيير الموقع الجغرافي

لو الـ VPN Server في أمريكا، المواقع هتحسب إنك في أمريكا. ده بيخليك تفتح محتوى محجوب في بلدك.

تجاوز الحجب

مواقع أو خدمات محجوبة في بلدك؟ الـ VPN بيقدر يتجاوز الحجب ده.

الـ VPN مش معناه إنك مجهول تماماً — لو سجّلت دخولك على موقع بحسابك، الموقع عارف مين أنت بصرف النظر عن الـ VPN.

امتى تستخدم VPN؟

◈ لما تكون على **WiFi عام** (كافيه، مطار) وعايز تحمي بياناتك

◈ لما تيجي تفتح **محتوى مقيد جغرافياً**

◈ لما عايز تحافظ على **خصوصيتك** من المواقع والـ ISP

◈ لما في **مواقع محجوبة** في بلدك عايز توصلها

Lesson 17

Internet -vs- WWW

ما هو الإنترنت؟

Internet: شبكة ضخمة جداً بتربط ملايين الشبكات والأجهزة ببعضها حول العالم كله — هي البنية التحتية اللي كل حاجة بتشتغل عليها , عشان كده بنسميها شبكة الشبكات.

احنا اتكلمنا قبل كده عن الـ LAN والـ WAN — الـ LAN شبكة صغيرة في البيت أو الشركة، والـ WAN شبكة بتربط شبكات LAN ببعضها على مسافات بعيدة.

الإنترنت هو أكبر WAN في التاريخ — هو شبكة بتربط مئات الملايين من شبكات الـ LAN والـ WAN حول العالم كله ببعضها، من كل البلدان، من كل القارات، في نفس الوقت.

طب إيه اللي بيربطهم فعلياً؟

الإنترنت مش بس سوفت وير أو فكرة — ده فيه هارد وير حقيقي جداً:

- **كابلات بحرية ضخمة** ممدودة في قاع المحيطات بتربط القارات ببعضها
- **سيرفرات** بالملايين منتشرة في data centers حول العالم
- **راوترات ضخمة** بتوجّه البيانات من بلد لبلد

يعني لما بتبعت رسالة لحد في أمريكا، الرسالة دي فعلياً بتتنقل جوه كابل ممدود في قاع المحيط الأطلنطي.

الإنترنت بيشيل إيه؟

الإنترنت هو الطريق اللي بيتنقل عليه **أي نوع من البيانات**:

- الإيميلات
- مكالمات الفيديو (Zoom, WhatsApp)
- الألعاب أونلاين
- المواقع والتطبيقات
- الملفات والصور
- وكل حاجة تانية

الإنترنت هو الـ infrastructure — هو الأوتوستراد العالمي اللي كل الخدمات بتتحرك عليه، هو نفسه مش خدمة، هو الوسيلة اللي الخدمات بتشتغل من خلالها.

ما هو الـ WWW؟

WWW: اختصار لـ World Wide Web — هو مجموعة المواقع والصفحات المترابطة ببعضها اللي بتقدر توصلها عن طريق المتصفح.

إزاي بيشتغل الـ WWW؟

الـ WWW قايم على **3 حاجات أساسية**:

- **HTTP/HTTPS** — البروتوكول اللي بيحدد إزاي المتصفح والسيرفر يتكلموا مع بعض
- **URL** — العنوان اللي بتكتبه في المتصفح عشان توصل لصفحة معينة (زي `www.google.com`)
- **HTML** — اللغة اللي بيتكتب بيها محتوى الصفحة اللي بتشوفها

لما بتكتب عنوان موقع في المتصفح، المتصفح بيبعت **Request** عن طريق الإنترنت للسيرفر بتاع الموقع ده، والسيرفر بيرد عليه بصفحة HTML تظهرلك على الشاشة.

الـ WWW بالمثال

تخيل الإنترنت زي **شبكة الطرق والأوتوسترادات** اللي بتربط كل المدن ببعضها.

الـ WWW هو **المحلات والمباني والمعالم** اللي على جنب الطريق ده — انت مش مسافر عشان تشوف الطريق، انت مسافر عشان توصل لمكان معين. الطريق (الإنترنت) هو الوسيلة، والمكان (الموقع) هو جزء من الـ WWW.

الفرق بينهم

	الإنترنت	WWW
هو إيه؟	شبكة اتصالات عالمية	مجموعة مواقع ومحتوى
الدور	ينقل البيانات	يعرض المحتوى
اخترع امتى؟	الستينات	1991
بيحتاج التاني؟	مستقل	محتاج الإنترنت
بتوصله إزاي؟	تلقائي لما تتصل	عن طريق المتصفح

ناس كتير بتقول "الإنترنت" وهما قاصدين "المواقع" — بس فعلياً المواقع هي الـ WWW، والإنترنت هو الشبكة اللي بتشيلها.

الـ WWW خدمة واحدة من ضمن خدمات كتير بتشتغل على الإنترنت — زي ما الواتساب خدمة والإيميل خدمة، الـ WWW كمان خدمة، بس الأشهر والأكبر.

Lesson 18

Browsers — المتصفحات

ما هو المتصفح؟

Browser: برنامج بيخليك تتصفح الإنترنت — بتكتب فيه عنوان موقع، وهو بيجيبهولك ويعرضهولك بشكل مفهوم.

أشهر المتصفحات اللي ممكن تكون شايفها:

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Safari

إيه اللي بيحصل لما بتكتب رابط؟

لما بتكتب عنوان موقع في المتصفح وتضغط Enter، المتصفح بيبعت Request للسيرفر بتاع الموقع ده.

السيرفر بيرد عليه بـ ملف — وده اللي المتصفح بيتعامل معاه.

طب الملف ده عبارة عن إيه؟

المتصفح فعلياً بيعمل Download لملف — اسمه **Webpage**.

الـ Webpage ده بيكون مكتوب بلغة اسمها HTML .

HTML: لغة وصفية (ليست لغة برمجة) بتوصف شكل ومحتوى الصفحة — بتقول فيها فين العنوان، فين الصورة، فين الزرار، وهكذا.

يعني الـ HTML مش بتشتغل — هي بس **بتوصف**. بتقول للمتصفح "في عنوان هنا، في صورة هناك، في نص في الوسط."

دور المتصفح

المتصفح بياخد كود الـ HTML ده، وبيترجمه ويحوله من كود لشكل مرئي تقدر تتعامل معاه.

المتصفح = محرك ترجمة — بياخد كود HTML مش مفهوم للإنسان، ويحوله لصفحة واضحة بصور

وألوان وأزرار.

مثال بسيط

تخيل إن الـ HTML كده:

```
<h1>أهلاً بيك</h1>
<p>ده موقعي الأول</p>
<img src="photo.jpg">
```

المتصفح بيقرا الكود ده ويعرضهولك كده:

أهلاً بيك

ده موقعي الأول

[صورة هنا]

نفس المحتوى — بس واحد كود، والتاني شكل مفهوم.

كل صفحة بتفتحها على الإنترنت هي فعلياً ملف HTML اتعمله Download على جهازك — المتصفح هو اللي بيعرضهولك بشكل جميل.

لاحظ ان الـ Webpage مش بس HTML , في الغالب بيتنزل معاها ملفات تانية زي CSS للتنسيق و JavaScript للتفاعل، بس HTML هي الأساس اللي بيبدأ منه كل حاجة.

Lesson 19

HTTP & HTTPS Protocols

في الدرس اللي فات اتكلمنا إن المتصفح بيبعت Request للسيرفر عشان يجيب الـ Webpage — طب بيبعت الـ Request ده بإيه؟ بيبعته عن طريق بروتوكول اسمه HTTP .

ما هو HTTP؟

HTTP: اختصار لـ HyperText Transfer Protocol — هو البروتوكول اللي بيحدد إزاي المتصفح والسيرفر يتكلموا مع بعض لنقل صفحات الويب.

لما بتكتب رابط وتضغط Enter، المتصفح بيبعت HTTP Request للسيرفر، والسيرفر يرد بـ HTTP Response فيها الـ Webpage.

المشكلة في HTTP إن البيانات بتتنقل من غير أي تشفير — يعني لو حد اتنصت على الاتصال هيقدر يقرا كل حاجة بتبعتها أو بتستقبلها بسهولة.

ما هو HTTPS؟

HTTPS: اختصار لـ HyperText Transfer Protocol Secure — هو HTTP بس بالتشفير والتحقق.

الـ HTTPS بيعمل نفس شغل الـ HTTP بالظبط، الفرق الوحيد إنه بيشفّر البيانات اللي بتتنقل بين المتصفح والسيرفر.

إزاي بيشفّر؟

الـ HTTPS بيستخدم بروتوكول تشفير اسمه TLS أو القديم منه SSL .

TLS: Transport Layer Security — البروتوكول المسؤول عن تشفير البيانات في الـ HTTPS.

SSL: Secured Socket Layer — النسخة القديمة من TLS، أبطأ وأقل أماناً، معظم المواقع تخلت عنه.

TLS هو الأحدث والأكثر أماناً وأسرع من SSL — المواقع الحديثة كلها بتستخدم TLS، بس لحد دلوقتي الناس كتير لسه بتقول "SSL" وهما قاصدين TLS.

الفرق بين HTTP و HTTPS

	HTTP	HTTPS
التشفير	❌ لا يوجد	✅ موجود (TLS)
الأمان	ضعيف	قوي
الـ Port	80	443
في الرابط	`http://`	`https://`
الاستخدام	نادر في المواقع الحديثة	المعيار الأساسي حالياً

إزاي تعرف الموقع بيستخدم إيه؟

بص على الرابط في المتصفح:

- ◈ https://🔒 → الاتصال مشفر وآمن
- ◈ http://⚠ → الاتصال مش مشفر

أي موقع بتحط فيه بياناتك الشخصية أو بيانات بنك — لازم يكون HTTPS مش HTTP، لأن HTTP بياناتك بتتنقل واضحة لأي حد يتنصت.

المتصفحات الحديثة زي Chrome بتحذرك تلقائياً لو دخلت موقع HTTP وبتعلم عليه بـ "Not Secure".

Lesson 20

What is Domain Name? From Where to Get One?

خليني أبدأ بسؤال

لما تكتب في المتصفح:

`facebook.com`

هل المتصفح بيفهم كلمة "facebook"؟

لا.

الكمبيوتر لا يفهم أسماء... الكمبيوتر يفهم أرقام فقط.

كل سيرفر على الإنترنت له **IP Address** .

زي ما شفنا:

◈ ProgrammingAdvices.com ↔ 104.19.238.117

◈ FaceBook.com ↔ 31.13.79.254

السيرفر له IP حقيقي. لكن نحن لا نحفظ IP... نحن نحفظ اسم.

ما هو الـ Domain؟

Domain Name: اسم سهل التذكر مرتبط بعنوان IP حقيقي على الإنترنت.

هو مجرد اسم يدل على سيرفر.

يعني:

◈ السيرفر موجود بعنوان رقمي

◈ الدومين يشير إليه باسم مفهوم للبشر

الدومين ليس هو الموقع. الدومين ليس هو السيرفر. هو فقط عنوان يدل على مكانهما.

أين يظهر الدومين؟

◈ بعد www في عنوان الموقع

◈ بعد @ في البريد الإلكتروني

مثال:

gmail.com `user@gmail.com` هو الدومين.

ماذا يحدث عندما تكتب Domain؟

عندما تكتب:

`facebook.com`

الذي يحدث هو:

1. المتصفح يسأل نظام اسمه DNS
2. DNS يترجم الاسم إلى IP
3. المتصفح يتصل بالسيرفر عبر الـ IP
4. السيرفر يرسل الصفحة

العملية كلها تتم في ثواني.

Part 2

From Where to Get One?

السؤال الآن:

كيف تحصل على Domain خاص بك؟

الدومين يتم تسجيله من خلال شركات اسمها:

Domain Registrars

أنت تختار اسم. إذا كان غير مستخدم، يمكنك تسجيله.

التسجيل لا يعني أنك تملكه للأبد. أنت تحجزه لمدة سنة أو أكثر. إذا لم تجدد، يمكن لشخص آخر تسجيله.

مكونات الدومين

مثال:

`mywebsite.com`

يتكون من:

◈ mywebsite ← الاسم الذي تختاره

◈ .com ← الامتداد (Top Level Domain)

أمثلة امتدادات:

- ◈ .com
- ◈ .net
- ◈ .org
- ◈ .eg

الدومين هو هوية رقمية لموقعك على الإنترنت.

كل موقع على الإنترنت يحتاج:

- ◈ سيرفر (له IP)
- ◈ دومين يشير إلى هذا السيرفر

الدومين هو الوسيط بين الإنسان والرقم.

Lesson 21

DNS — Domain Name Server

إحنا اتفقنا إن:

- ◈ كل سيرفر له **IP Address**
- ◈ البشر يستخدمون **Domain Name**

السؤال الآن:

كيف يتحول الاسم إلى رقم؟

هنا يأتي دور الـ **DNS** .

ما هو DNS؟

DNS يشبه دليل الهاتف.

في دليل الهاتف:

- ◈ تبحث بالاسم
- ◈ تحصل على الرقم

في DNS:

- ◈ تبحث باسم الموقع
- ◈ تحصل على الـ IP

DNS (Domain Name Server): نظام مسؤول عن تحويل اسم الدومين إلى عنوان IP.

بدونه... الإنترنت لا يعمل بالأسماء.

ماذا يحدث عند كتابة domain في المتصفح؟

لنفرض أنك كتبت:

`google.com`

الخطوات تكون كالتالي:

1. المتصفح لا يعرف الـ IP
2. يسأل DNS: ما هو IP الخاص بـ google.com؟
3. DNS يرد بالـ IP الصحيح
4. المتصفح يتصل بالسيرفر باستخدام الـ IP

5. السيرفر يرسل الصفحة

كل هذا يتم خلال أجزاء من الثانية.

أين يوجد DNS؟

DNS ليس سيرفر واحد فقط.

هو نظام عالمي موزع على آلاف السيرفرات حول العالم.

عندما تسأل عن IP:

الطلب لا يذهب لسيرفر واحد فقط بل يمر بعدة مراحل حتى يصل للإجابة الصحيحة.

لماذا لا نحفظ IP مباشرة وننتهي؟

لأن:

- IP يمكن أن يتغير
- السيرفرات قد تنتقل لمكان آخر
- الشركة قد تغير مزود الخدمة

لو كنت تحفظ IP مباشرة كل مرة يتغير فيها IP سيتوقف الموقع بالنسبة لك

لكن باستخدام DNS:

يمكن تغيير IP في الخلفية ويبقى اسم الدومين ثابتاً

وهذه نقطة مهمة جداً.

مثال بسيط

لنفترض:

facebook.com كان مربوط بـ IP معين

إذا قامت الشركة بتغيير السيرفر يتغير IP

المستخدم لا يشعر بأي شيء لأن الاسم لم يتغير وDNS فقط يتم تحديثه

DNS هو الوسيط الذكي بين الاسم الذي يفهمه الإنسان والرقم الذي يفهمه الكمبيوتر.

Lesson 22

Sub Domain

عندنا شيء اسمه **Domain Name** مثل:

- google.com
- facebook.com

هذا هو العنوان الرئيسي للموقع.

لكن هل الموقع الكبير يكون صفحة واحدة فقط؟ لا.

أي موقع كبير يحتوي على أقسام كثيرة:

- متجر
- مدونة
- لوحة تحكم
- دولة معينة
- إدارة

كيف نفصل هذه الأقسام بدون شراء دومين جديد كل مرة؟

هنا يظهر مفهوم:

Sub Domain

ما هو الـ Subdomain؟

Subdomain: جزء إضافي يتم وضعه قبل الدومين الأساسي لإنشاء قسم أو موقع فرعي منه.

السب دومين هو جزء إضافي يوضع في بداية اسم الدومين لتنظيم محتوى الموقع

الشكل العام يكون:

```
subdomain.domain.com
```

أمثلة مباشرة

لدينا الدومين الأساسي: **programmingadvices.com**

يمكن إنشاء:

- ◈ blog. programmingadvices.com
- ◈ management. programmingadvices.com
- ◈ usa. programmingadvices.com
- ◈ jordan. programmingadvices.com

هذه كلها Subdomains لنفس الموقع الأساسي — كل واحد يمثل قسم مختلف.

لماذا نستخدم Subdomain؟

لتقسيم الموقع الكبير إلى أجزاء واضحة. بدلاً من أن يكون الموقع كله مختلطاً، يتم فصل كل وظيفة في Subdomain مستقل.

مثال — موقع شركة:

الـ Subdomain	الوظيفة
`www.company.com`	الموقع الرئيسي
`shop.company.com`	المتجر
`blog.company.com`	المدونة
`admin.company.com`	لوحة التحكم

كل جزء منفصل لكنه تابع لنفس الدومين.

هل الـ Subdomain يحتاج شراء؟

لا. عند امتلاكك للدومين الأساسي يمكنك إنشاء عدد غير محدود من Subdomains مجاناً، لأنه تابع للدومين الأساسي.

هل الـ Subdomain موقع مستقل؟

تقنياً نعم. يمكن ربط كل Subdomain بـ:

- ◈ سيرفر مختلف
- ◈ تطبيق مختلف
- ◈ نظام مختلف

يعني: `blog.example.com` يمكن أن يكون على سيرفر مختلف تماماً عن `shop.example.com` — لكن كلاهما تابع لنفس الدومين.

التركيب

مثال: `blog.google.com`

الجزء	النوع
blog	Subdomain
google	Domain Name
com.	TLD

الدومين هو العنوان الرئيسي — والـ Subdomain هو تقسيم داخلي للموقع.

Lesson 23

URL — Uniform Resource Locator

أي شيء تفتحه على الإنترنت له عنوان. هذا العنوان يسمى: URL

الـ URL هو العنوان الكامل لأي مورد على الشبكة.

المورد قد يكون:

- ◈ صفحة ويب
- ◈ صورة
- ◈ ملف
- ◈ فيديو
- ◈ API
- ◈ أي شيء موجود على سيرفر

ما معنى URL؟

URL: عنوان يشير إلى مورد على الشبكة ويحدد مكانه وطريقة الوصول إليه.

هو لا يحدد المكان فقط، بل يحدد أيضاً الطريقة المستخدمة للوصول — يعني: أين يوجد الشيء وكيف أصل إليه.

مثال مباشر

عندما تكتب: `https://www.google.com`

هذا يسمى URL كامل. وعندما تضغط Enter، المتصفح يفهم منه كل المعلومات اللازمة للوصول للموقع.

مكونات الـ URL

مثال:

```
https://www.google.com/index.html
```

1) Protocol (البروتوكول)

`https://` — هو الطريقة المستخدمة للاتصال بالسيرفر.

أشهرها:

- **http**
- **https** — الأكثر استخداماً لأنه آمن ومشفر
- **ftp**

2) Domain Name

`www.google.com` — هو اسم الموقع الذي يشير إلى السيرفر. يتم تحويله إلى IP بواسطة DNS ثم يتم الاتصال بالسيرفر.

3) Path (المسار)

`/index.html` — يحدد الصفحة أو الملف داخل السيرفر بالضبط.

الفكرة العامة

الـ URL يشبه العنوان البريدي الكامل:

العنوان البريدي	مكوّن الـ URL
طريقة الوصول	البروتوكول
اسم البلد والمدينة	الدومين
الشارع ورقم الشقة	المسار (Path)

هل كل صفحة لها URL مختلف؟

نعم. كل صفحة أو صورة أو ملف له URL خاص به. مثال:

- `google.com`
- `google.com/maps`
- `google.com/mail`
- `google.com/images`

كل واحد منهم مورد مختلف وله URL مختلف.

الـ URL هو العنوان الكامل لأي مورد على الإنترنت، ويحدد مكانه وطريقة الوصول إليه.

Lesson 24

FTP — File Transfer Protocol

لنفترض أن عندك موقع على الإنترنت وعندك ملفات:

- صور
- فيديوهات
- ملفات HTML
- ملفات CSS
- ملفات مشروع

كيف ترفع هذه الملفات إلى السيرفر؟ وكيف تنزل ملفات من السيرفر إلى جهازك؟

هنا يظهر بروتوكول مهم جداً:

FTP

ما هو الـ FTP؟

FTP: اختصار لـ File Transfer Protocol — أي بروتوكول نقل الملفات.

وظيفته:

- رفع الملفات إلى السيرفر **Upload**
- تحميل الملفات من السيرفر **Download**
- نقل الملفات بين أجهزة مختلفة

لماذا FTP تحديداً؟

أي نقل ملفات عبر الإنترنت يحتاج طريقة محددة. مثلما أن:

البروتوكول	الوظيفة
HTTP	نقل صفحات الويب
SMTP	إرسال الإيميل
FTP	نقل الملفات

هو البروتوكول المسؤول عن نقل الملفات فقط.

أين يُستخدم FTP؟

- ◈ رفع ملفات المواقع إلى السيرفر
- ◈ إدارة ملفات الاستضافة
- ◈ نقل ملفات كبيرة بين الأجهزة
- ◈ نسخ احتياطية على السيرفر

أي مطور مواقع يستخدم FTP باستمرار.

كيف تستخدم FTP؟

لكي تستخدم FTP تحتاج:

1. **سيرفر يدعم FTP**
2. **برنامج FTP على جهازك**
3. **بيانات الدخول للسيرفر**

عند الاتصال:

جهازك ↔ سيرفر FTP

يمكنك:

- ◈ رفع ملفات
- ◈ حذف ملفات

◈ تعديل ملفات

◈ تنزيل ملفات

وكأنك تتعامل مع ملفات على جهازك.

مميزات FTP

---

1) نقل ملفات كثيرة مرة واحدة

يمكن رفع مجلد كامل يحتوي آلاف الملفات دفعة واحدة.

2) استكمال التحميل عند الانقطاع

إذا انقطع الإنترنت أثناء النقل يمكن إكماله من نفس المكان.

3) جدولة النقل

يمكن تحديد وقت معين لرفع أو تنزيل الملفات.

4) لا يوجد حد لحجم الملف

المتصفح غالباً يحدد حجم معين للرفع، لكن FTP لا يضع حد عملياً — يمكن نقل ملفات ضخمة جداً.

5) رفع وتنزيل في Queue

يمكن وضع ملفات كثيرة في قائمة ويتم رفعها أو تنزيلها تلقائياً.

مثال عملي

عند إنشاء موقع: المبرمج يكتب الموقع على جهازه، ثم يستخدم FTP لرفع الملفات إلى السيرفر. بعد الرفع، الموقع يظهر على الإنترنت. بدون FTP لن يستطيع نقل ملفات الموقع بسهولة.

Lesson 25

What is API?

لما بتفتح تطبيق الجو وبتشوف درجة الحرارة، لما بتدفع أونلاين بـ Visa، لما بتسجل دخول بحساب Google — كل ده بيحصل عن طريق API

ما هو الـ API؟

API: اختصار لـ Application Programming Interface — هو الطريقة التي يتكلم بيها برنامج مع برنامج

تاني.

تخيل كده...

انت في مطعم.

◈ انت مش بتروح للمطبخ وتطبخ بنفسك.

◈ انت بتطلب من **الويتر**

◈ الويتر بياخد طلبك، يروح للمطبخ، ويرجعلك الأكل.

الـ API هو الويتر — الوسيط بين انت والسيرفر.

انت = التطبيق

الويتر = API

المطبخ = السيرفر أو قاعدة البيانات

مثال حقيقي — تطبيق الطقس

عندك تطبيق جو على تليفونك.

التطبيق ده مش عنده بيانات الطقس بنفسه. فبيعمل إيه؟

الخطوة	اللي بيحصل
1	التطبيق بيبعت طلب لـ API
2	الـ API بيروح لسيرفر بيانات الطقس
3	السيرفر بيرد بالبيانات
4	الـ API بيرجع البيانات للتطبيق
النتيجة	بتشوف درجة الحرارة على شاشتك

التطبيق ما اتعاملش مع السيرفر مباشرة — الـ API هو اللي اتوسط.

مثال تاني — الدفع أونلاين

لما بتشتري من موقع وبتدفع بـ Visa:

◈ الموقع مش بنفسه بيتعامل مع البنك.

◈ بيستخدم **Payment API** زي Stripe أو PayPal.

◈ الـ API بيتكلم مع البنك، بيتأكد من الرصيد، وبيرجع النتيجة.

الموقع ما عرفش أي بيانات عن كارتك — كل ده اتعمل جوه الـ API.

ليه بنستخدم API؟

---

1) مش لازم تبني كل حاجة من الصفر

عايز تضيف خرايط في موقعك؟ استخدم **Google Maps API** — جاهز وشغال.

2) الأمان

التطبيق بيوصل للبيانات اللي محتاجها بس — مش أكتر.

3) سهولة التواصل بين الأنظمة

تطبيق موبايل + موقع ويب + برنامج Desktop — كلهم يتكلموا مع نفس الـ API.

4) توفير الوقت والجهد

بدل ما تبني نظام دفع أو نظام إيميل أو نظام خرايط — تستخدم API جاهز.

شكل الـ API في الحقيقة

الـ API عبارة عن **رابط (URL)** بتبعته طلب ويرد عليك ببيانات.

مثال:

```
https://api.weather.com/cairo
```

بتبعته طلب ← يرد عليك بـ:

```
{
  "city": "Cairo",
  "temp": 28,
  "status": "Sunny"
}
```

البيانات بترجع في الغالب بصيغة JSON — وهي صيغة بسيطة وسهلة القراءة للبرامج.

Lesson 26

What is XML?

تخيل إنك عندك بيانات عايز تبعتها من برنامج لبرنامج تاني.

المشكلة: كل برنامج بيتكلم بطريقته الخاصة. فكيف يفهم كل برنامج البيانات اللي جاتله؟

الحل:

XML

ما هو الـ XML؟

XML: اختصار لـ eXtensible Markup Language — لغة لتنظيم البيانات وتخزينها بشكل واضح يقدر أي برنامج يقراه.

XML مش لغة برمجة — هي طريقة لتنظيم البيانات فقط.

ماهية الـ XML

الـ XML ببساطة هو:

◈ **وعاء للبيانات** — بيحتوي المعلومات بشكل منظم.

◈ **مستقل عن أي برنامج** — أي لغة برمجة تقدر تقراه.

◈ **مقروء للإنسان والآلة** — واضح للمبرمج وللبرنامج في نفس الوقت.

الفكرة الأساسية: البيانات بتتحط جوه **Tags** زي HTML تماماً — لكن الفرق إن XML انت اللي بتسمي الـ Tags بنفسك على حسب البيانات.

مثال واضح

عندك بيانات موظف — كيف تكتبها في XML؟

```
<employee>
    <name>Ahmed Ali</name>
    <age>28</age>
    <job>Developer</job>
    <salary>5000</salary>
</employee>
```

كل بيانة محاطة بـ Tag بيوصف محتواها — `<name>` جوه `</name>` يعني ده اسم. `<age>` جوه `</age>` يعني دي سن.

واضح ومنظم تماماً.

مثال أكبر — قائمة موظفين

```
<employees>

    <employee>
        <name>Ahmed Ali</name>
        <job>Developer</job>
        <salary>5000</salary>
    </employee>

    <employee>
        <name>Sara Mohamed</name>
        <job>Designer</job>
        <salary>4500</salary>
    </employee>

</employees>
```

لاحظ إن الـ Tags متداخلة — كل موظف جوه `employees`، وبيانات كل موظف جوه `employee`. ده اللي بيخلي البيانات منظمة وهرمية.

XML مقارنةً بـ JSON

الاتنين بيعملوا نفس الشيء — تنظيم البيانات. لكن:

	XML	JSON
الشكل	Tags زي HTML	أبسط وأخف
الاستخدام	أنظمة قديمة وـ APIs كتير	الأشهر في الـ APIs الحديثة
القراءة	أوضح للإنسان	أسرع للبرنامج

أين يُستخدم XML؟

- ◈ تبادل البيانات بين الأنظمة المختلفة
- ◈ ملفات الإعدادات في البرامج `config.`
- ◈ الـ APIs القديمة `SOAP`
- ◈ ملفات Office زي Excelg Word من جوا `docx / .xlsx.`

الخلاصة: XML هو لغة لتنظيم البيانات داخل Tags — بتخلي أي برنامج يقدر يقرأ البيانات ويفهمها

بغض النظر عن اللغة أو النظام.

Lesson 27

What is JSON?

في الدرس اللي فات اتكلمنا عن XML كطريقة لتنظيم البيانات ونقلها بين البرامج عن طريق Tags — الـ JSON بيعمل نفس الفكرة بالظبط، بس بطريقة مختلفة في الشكل والأسلوب.

ما هو JSON؟

JSON: اختصار لـ JavaScript Object Notation — طريقة لتنظيم البيانات وتخزينها بشكل بسيط وخفيف يقدر أي برنامج يقراه.

زي الـ XML تماماً — JSON مش لغة برمجة، هي بس طريقة لتنظيم البيانات.

شكل الـ JSON

البيانات في JSON بتتكتب على شكل Key : Value — يعني اسم البيانة، ثم نقطتين، ثم قيمتها.

نفس مثال الموظف اللي شفناه في XML:

```
{
    "name": "Ahmed Ali",
    "age": 28,
    "job": "Developer",
    "salary": 5000
}
```

الـ Keys دايماً بين " " — والـ Values ممكن تكون نص بين " "، أو رقم بدون " "، أو قيم تانية هنتكلم عنها.

مثال أكبر — قائمة موظفين

لو عندنا أكتر من موظف، بنحطهم جوه Array باستخدام [ ] :

```
{
    "employees": [
        {
            "name": "Ahmed Ali",
            "job": "Developer",
            "salary": 5000
        },
        {
            "name": "Sara Mohamed",
            "job": "Designer",
            "salary": 4500
        }
    ]
}
```

الـ [ ] دي بتقول "في أكتر من عنصر" — زي قائمة. وكل عنصر جواها بين { }.

الفرق بين JSON و XML

	XML	JSON
الشكل	Tags لكل بيانة فتح وغلق	Key : Value بدون Tags
الحجم	أكبر وأثقل	أصغر وأخف
السرعة	أبطأ في المعالجة	أسرع
القراءة	أوضح للإنسان	أبسط وأسهل
الاستخدام	الأنظمة القديمة و SOAP APIs	المعيار في الـ APIs الحديثة

نفس البيانات — فرق الشكل

```
<employee>
  <name>Ahmed Ali</name>
  <age>28</age>
  <job>Developer</job>
</employee>
```

```
{
  "name": "Ahmed Ali",
  "age": 28,
  "job": "Developer"
}
```

نفس البيانات بالظبط — بس JSON أقل كتابة وأخف حجماً.

امتى تستخدم إيه؟

- ◈ **JSON** → الـ APIs الحديثة، تطبيقات الويب والموبايل، أي مشروع جديد
- ◈ **XML** → الأنظمة القديمة، ملفات الإعدادات، SOAP APIs، ملفات Office

الـ JSON أصبح المعيار الأساسي في الـ APIs الحديثة — لو بتبني أي تطبيق جديد هتتعامل مع JSON أكتر بكتير من XML.

Lesson 28

What is GUID?

ما هو الـ GUID؟

GUID: اختصار لـ Globally Unique Identifier — رقم تعريفي فريد من نوعه على مستوى العالم كله، بيتولد تلقائياً ومستحيل يتكرر.

شكله

الـ GUID بيتكتب كده:

```
550e8400-e29b-41d4-a716-446655440000
```

بيتكون من 32 خانة هكساديسمال مقسمة بـ 4 شرطات على شكل 5 مجموعات:

`550e8400` - `e29b` - `41d4` - `a716` - `446655440000`

8 - 4 - 4 - 4 - 12 خانة

ليه محتاجين GUID؟

تخيل إنك بتبني تطبيق وعندك جدول مستخدمين — كل مستخدم محتاج ID مميز يعرفه.

الطريقة العادية إنك تستخدم أرقام تسلسلية: 1، 2، 3، 4...

المشكلة في الأرقام التسلسلية:
— لو عندك أكتر من Database أو أكتر من سيرفر، الأرقام هتتعارض.
— لو حذفت record ورقمه اتعمل مرة تانية، هيبقى في تعارض في البيانات القديمة.
— سهل تخمن IDs المستخدمين التانيين.

الـ GUID بيحل كل المشاكل دي — لأنه فريد على مستوى العالم كله، مش بس على مستوى الـ Database بتاعتك.

فايدته

- **فريد عالمياً** — مستحيل يتكرر نفس الـ GUID في أي مكان في العالم
- **مش قابل للتخمين** — أكثر أماناً من الأرقام التسلسلية
- **مناسب للأنظمة الموزعة** — لو عندك أكتر من سيرفر أو Database مفيش تعارض
- **مستقل** — تقدر تولّده من غير ما ترجع للـ Database خالص

مثال بالـ #C

```
using System;

// توليد GUID جديد
Guid userId = Guid.NewGuid();
Console.WriteLine(userId);
// Output: 550e8400-e29b-41d4-a716-446655440000

// مقارنة بين GUID مختلفين — دايماً اتنين
Guid id1 = Guid.NewGuid();
Guid id2 = Guid.NewGuid();
Console.WriteLine(id1 == id2); // False
```

مش بس #C — أي لغة برمجة

الـ GUID مفهوم عالمي مش مرتبط بلغة معينة — كل لغة عندها طريقتها:

اللغة	الكود
#C	`Guid.NewGuid()`
Python	`import uuid` ثم `uuid.uuid4()`
JavaScript	`crypto.randomUUID()`
Java	`UUID.randomUUID()`
SQL	`NEWID()`

> **الـ GUID بيتستخدم كثيراً في قواعد البيانات كـ Primary Key، في الـ APIs كـ ID للـ Requests، وفي الأنظمة الكبيرة الموزعة على أكتر من سيرفر.**

Lesson 29

What is 3-Tier Architecture?

تخيل إنك بتبني تطبيق وكتبت كل حاجة في مكان واحد — الـ UI والمنطق وقاعدة البيانات كلهم مع بعض في نفس الملف.

لو عايز تغير أي حاجة بسيطة، هتلمس كل الكود وممكن تكسر أي جزء من التطبيق — وكل ما التطبيق كبر، كل ما الأمور اتعقدت أكتر.

الحل إنك تقسم التطبيق لـ **طبقات** كل واحدة مسؤولة عن حاجة معينة — وده اللي بيعمله الـ **3-Tier Architecture**.

ما هو الـ 3-Tier Architecture؟

3-Tier Architecture: طريقة لتصميم التطبيقات بتقسمه لـ 3 طبقات منفصلة — كل طبقة ليها مهمة واحدة بس، ومش بتتكلم إلا مع اللي جنبها.

الـ 3 طبقات هم:

الواجهة — Presentation Tier

(من و إلى)

المنطق — Business Logic Tier

(من و إلى)

البيانات — Data Tier

الطبقة الأولى — Presentation Tier

Presentation Tier: الجزء اللي المستخدم بيشوفه ويتعامل معاه — الـ UI بتاع التطبيق.

- ◈ هي اللي بتعرض البيانات للمستخدم
- ◈ بتستقبل منه الـ Input
- ◈ مش بتعمل أي منطق أو حسابات — بس بتعرض وتستقبل
- ◈ أمثلة: صفحة ويب، شاشة موبايل، Desktop App

مثال

صفحة فيها Form المستخدم بيكتب فيها اسمه وباسورده ويضغط Login — الصفحة دي هي الـ Presentation Tier، هي بس بتاخد الداتا وتبعتها للطبقة اللي تحتها.

الطبقة التانية — Business Logic Tier

Business Logic Tier: الجزء اللي فيه المنطق الحقيقي للتطبيق — بيقرر إيه اللي هيحصل بناءً على الـ Input الجاي من الـ UI.

- ◈ هي العقل بتاع التطبيق
- ◈ بتاخد الداتا من الـ Presentation، تعالجها، وتطلب من الـ Data Tier البيانات اللي محتاجاها
- ◈ كل قواعد التطبيق موجودة هنا

مثال

لما المستخدم ضغط Login، الـ Business Logic هي اللي بتقول: "خد الاسم والباسورد، روح تحقق منهم في قاعدة البيانات، لو صح سمحله يدخل، لو غلط قوله Error."

الطبقة التالتة — Data Tier

Data Tier: الجزء اللي بيخزن البيانات ويرجعها — قاعدة البيانات.

- ◈ مسؤولة بس عن تخزين واسترجاع البيانات
- ◈ مش بتعرف حاجة عن الـ UI أو المنطق
- ◈ أمثلة: SQL Server, MySQL, MongoDB

مثال

الـ Business Logic طلبت "جبلي بيانات المستخدم اللي اسمه Ahmed" — الـ Data Tier بتروح في قاعدة البيانات وبترجع البيانات دي بس، من غير ما تعرف هتتعمل فيها إيه.

الصورة الكاملة — مثال Login

الخطوة	الطبقة	اللي بيحصل
1	Presentation	المستخدم كتب الاسم والباسورد وضغط Login
2	Business Logic	استقبلت البيانات وطلبت من الـ DB تتحقق منها
3	Data	دورت في قاعدة البيانات ورجعت النتيجة
4	Business Logic	حللت النتيجة وقررت "يدخل" أو "يترفض"
5	Presentation	عرضت للمستخدم "أهلاً بيك" أو "باسورد غلط"

ليه الـ Tier-3 أحسن؟

- ◈ **الفصل** — كل طبقة شغلانة في حاجتها بس، مش حد شايل هم التاني
- ◈ **سهولة التعديل** — عايز تغير شكل الـ UI؟ غير الـ Presentation بس من غير ما تلمس المنطق أو البيانات
- ◈ **سهولة الـ Testing** — تقدر تتيست كل طبقة لوحدها
- ◈ **الـ Scalability** — تقدر تكبّر أي طبقة لوحدها لو احتجت

الـ Tier-3 مش بس نظرية — ده اللي بيحصل فعلاً في كل تطبيق ويب حديث. الـ Frontend هو الـ Presentation، الـ Backend هو الـ Business Logic، وقاعدة البيانات هي الـ Data Tier.

الطبقات بتتكلم مع بعضها بالترتيب بس — الـ Presentation مش بتتكلم مع الـ Data مباشرة أبداً، لازم تعدي على الـ Business Logic الأول.