بسم الله الرحمن الرحيم

السلام عليكم و رحمة الله تعالى و بركاته

في هذا الدرس بإذن الله سوف نتطرق إلى عدد جديد من المواضيع الشيقة التي لا غنى عنها لأي مبرمج مهما كان تخصصه

لذا نبدأ على بركة الله .

1 – أنظمة ترميز الحروف و الرموز

منذ ظهور الحاسوب و مشكلة الترميز تعتبر من القضايا الشائكة التي حيرت المبرمجين و مصنعي الحاسوب فربما كانت الشركة الواحدة تملك أكثر من نظام ترميز لمختلف الاجهزة ، طبعا هذا الامر حال دون وجود اتصال بين مختلف انواع الحواسيب فكل شركة كانت تضع ترميز يختلف عن الأخرى ، إلى أن جاء أحد العاملين في شركة IBM عام 1960 و هو "Bob Bamer" و قام بالعمل على أول نظام ترميز تعمل به جميع أنواع الحواسيب و هو (**ASCII** - **American Standard Code for Information Interchange**) ، نتعرف هنا على أشهر أنظمة الترميز التي قد تمر عليك :

1 - **Extended Binary Coded Decimal Interchange Code - EBCDIC :** ظهر هذا النظام من قبل IBM الاولى ، و كان يعتمد فقط على 6 بت . و كان شائع الاستعمال في اجهزة

TABLE 6
EBCDIC (IBM MAINFRAME) CHARACTER CODES

Each code is shown in decimal, hexadecimal, and character form.

129	81	a	193	C1	A	240	F0	0
130	82	b	194	C2	B	241	F1	1
131	83	c	195	C3	C	242	F2	2
132	84	d	196	C4	D	243	F3	3
133	85	e	197	C5	E	244	F4	4
134	86	f	198	C6	F	245	F5	5
135	87	g	199	C7	G	246	F6	6
136	88	h	200	C8	H	247	F7	7
137	89	i	201	C9	I	248	F8	8
						249	F9	9
145	91	j	209	D1	J			
146	92	k	210	D2	K	64	40	blank
147	93	l	211	D3	L	76	4C	<
148	94	m	212	D4	M	77	4D	(
149	95	n	213	D5	N	78	4E	+
150	96	o	214	D6	O	79	4F	|
151	97	p	215	D7	P	80	50	&
152	98	q	216	D8	Q	90	5A	!
153	99	r	217	D9	R	91	5B	$
						92	5C	*
162	A2	s	226	E2	S	93	5D	)
163	A3	t	227	E3	T	94	5E	;
164	A4	u	228	E4	U	96	60	-
165	A5	v	229	E5	V	97	61	/
166	A6	w	230	E6	W	107	6B	,
167	A7	x	231	E7	X	108	6C	%
168	A8	y	232	E8	Y	109	6D	_
169	A9	z	233	E9	Z	110	6E	>
						111	6F	?
122	7A	:	125	7D	'			
123	7B	#	126	7E	=			
124	7C	@	127	7F	"			

2 - **ASCII :** كما قلنا سابقا يعتبر أول نظام ترميز للحواسيب يتألف من 7 بت يخزن فيها 128 حرف و رمز و هي تشمل حروف الابجدية الانجليزية (الأميركية) بشكليها الصغير و الكبير ، جميع علامات الترقيم مثل ، : . .....الخ

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☻	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	◘	BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♫	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	►	DLE	048	0	080	P	112	p
017	◄	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	‼	DC3	051	3	083	S	115	s
020	¶	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↨	ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(cursor right)	FS	060	<	092	\	124	|
029	(cursor left)	GS	061	=	093	]	125	}
030	(cursor up)	RS	062	>	094	^	126	~
031	(cursor down)	US	063	?	095	_	127	⌂

3 - **PASCII (Perso-Arabic Script Code for Information Interchange) :** و هو يعتبر أول نظام ترميز للغة العربية في الحواسيب صنعته الهن و كان يستعمل في التعاملات في أقليم كاشمير و أغلب الدول الناطقة بالعربية في آسيا

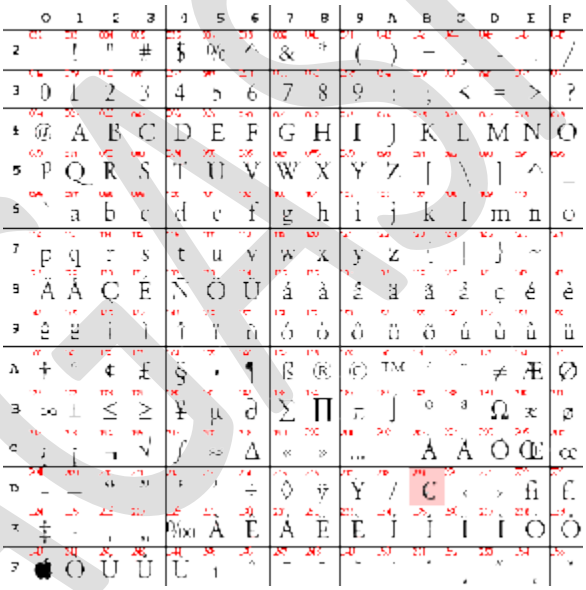
4 - **ArmSCII :** و هو الجدول الخاص بحروف اللغة الأرمنية .

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
0	NUL	DLE	SP	0	@	P	`	p	Ա	ՠ	ՠ				Ո	Ր
1	SOH	DC1	!	1	A	Q	a	q	ա	բ	ձ				ո	ր
2	STX	DC2	"	2	B	R	b	r	բ	ց	ղ				ջ	ց
3	ETX	DC3	#	3	C	S	c	s	գ	դ	ճ				ւ	գ
4	EOT	DC4	$	4	D	T	d	t	դ	ե	մ				պ	ի
5	ENQ	NAK	%	5	E	U	e	u	ե	զ	յ				ա	ւ
6	ACK	SYN	&	6	F	V	f	v	զ	է	ն				ջ	փ
7	BEL	ETB	'	7	G	W	g	w	է	ը	շ				ջ	փ
8	BS	CAN	(	8	H	X	h	x	ը	թ	ո				ի	ֆ
9	HT	EM	)	9	I	Y	i	y	թ	ժ	չ				ռ	ք
A	LF	SUB	*	:	J	Z	j	z	ջ	ի	պ				ս	օ
B	VT	ESC	+	;	K	[	k	{	ի	լ	ջ				ս	օ
C	FF	FS	,	<	L	\	l	|	խ	կ	ռ			❁	վ	ֆ
D	CR	GS	-	=	M	]	m	}	ծ	կ	ղ			~	վ	ֆ
E	SO	RS	.	>	N	^	n	~	ղ	հ	«			…	տ	’
F	SI	US	/	?	O	_	o	DEL	ղ	ֆ	»			՞	տ	’

5 – **Extended Unix Code) EUC (:** هو نظام ترميز يستخدم في بلاد شرق آسيا (الصين و كوريا و اليابان).

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x8FF4A0		Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ	Ⅹ	ˊ	ˋ	㈱	No.	℡	炻	任
0x8FF4B0	傰	凬	匆	匡	夔	咏	垔	垥	塚	增	寬	峐	嵜	崎	德	悅
0x8FF4C0	愠	敎	昂	皖	晴	朗	柳	椗	樥	橫	欟	淸	滬	瀨	熙	猪
0x8FF4D0	瓶	皂	皞	益	礪	礼	神	祥	福	竦	靖	精	綠	緒	羽	荲
0x8FF4E0	藹	薰	蘢	蛙	蠣	諸	讌	賴	赶	赳	迩	逸	郎	都	鄉	錚
0x8FF4F0	鋅	閒	隆	陽	靈	靨	靑	飯	飼	館	醇	高	鮏	鶴	黑	

6 – **نظام الترميز الخاص بأجهزة ماك (Mac OS code pages) :** هي أيضا مجموعة من الجداول التي إحتوت مجموعات مختلفة من الحروف و الرموز و كان من اللغات التي دعمتها (العربية ، الرومانية ، الاوكرانية) .



7 - **صفحات الترميز الخاصة بالدوس (DOS code pages) :** و هي مجموعة أيضا من الجداول تحتوي ترميزات الحروف و الاعداد لكن المميز فيها أنها كانت مبنية على ASCII لذا فهي كانت أكثر انتشارا .

8 - **صفحات الترميز الخاصة بنظام ويندوز (ANSI) :** هي مجموعة كبيرة من الصفحات التي أنشأتها شركة مايكروسوفت لنظام التشغيل ويندوز ، لم تعد الشركة تعمل بها حاليا ألا أنه توجد بعض الأجهزة (خاصة الصينية) تستعمل هذه الانظمة القديمة ، من أهم الصفحات بالنسبة لنا هو 1256 حيث أنها تحتوي على ترميز اللغة العربية .

9 – **ATARI ASCII _ - ATASCII :** هو نظام الترميز الخاص بجهاز الأتاري (مشغل الألعاب) و يتعتبر هو أول نظام ترميز يصمم لجهاز ألعاب .

ATASCII

00	♥	10	♣	20		30	0	40	@	50	P	60	♦	70	p
01	├	11	┌	21	!	31	1	41	A	51	Q	61	a	71	q
02	│	12	─	22	"	32	2	42	B	52	R	62	b	72	r
03	┘	13	┼	23	#	33	3	43	C	53	S	63	c	73	s
04	┤	14	●	24	$	34	4	44	D	54	T	64	d	74	t
05	┐	15	▬	25	%	35	5	45	E	55	U	65	e	75	u
06	╱	16	▕	26	&	36	6	46	F	56	V	66	f	76	v
07	╲	17	┬	27	'	37	7	47	G	57	W	67	g	77	w
08	◢	18	┴	28	(	38	8	48	H	58	X	68	h	78	x
09	▝	19	▌	29	)	39	9	49	I	59	Y	69	i	79	y
0A	◣	1A	└	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
0B	▘	1B	╭	2B	+	3B	;	4B	K	5B	[	6B	k	7B	♠
0C	▀	1C	↑	2C	,	3C	<	4C	L	5C	\	6C	l	7C	|
0D	▁	1D	↓	2D	-	3D	=	4D	M	5D	]	6D	m	7D	↰
0E	▂	1E	←	2E	.	3E	>	4E	N	5E	^	6E	n	7E	◂
0F	▃	1F	→	2F	/	3F	?	4F	O	5F	_	6F	o	7F	▸

10 - **نظام الترمز العالمي (Unicode transformation format 8 bit - UTF-8) :** يعتبر هذا هو الحل الأمثل لجميع مشاكل الترميز حول العالم فهذا المرجع الدولي يضم جميع لغات العالم مستخدما 8 بت .

2 – إتفاقية الترخيص

أثناء مسيرتك المهنية و مع تطور مشاريعك ستجد نفسك في حاجة ملحة لوضع ضامن لحقوقك الفكرية و وثيقة تأكد ملكيتك لهذا البرنامج ، هذه الوثيقة تسمى " إتفاقية الترخيص" و التي تعني أنك أعطيت إذن للمستخدم باستعمال برنامجك ، ففالماضي الحواسيب لم تكن مزودة ببرامج بل كان على المستخدم ان يكتب برامجه بنفسه ، كان الجميع مبرمجين ☺

بعيدا عن هذه المقدمة ، كيف لي أن أضع إتفاقية ترخيص ؟؟؟

للإجابة عن هذا السؤال سنستعرض معا عدد من الحلول ، أنت بدورك يمكنك أن تختار الأفضل بالنسبة لك .

1 - الاستعانة بمحامي : في حالة كان برنامجك أو تطبيقك تجاري و سعره عالي ، زبائنك من الشركات اذا يستفضل في هذه الحالة أن تعد لنفسك إتفاقية ترخيص خاصة بك تضم كافة الشروط و البنود التي تلائمك .

2 - GPL - **GNU General Public License** : ان كان برنامجك مجاني و مفتوح المصدر فلا يوجد أفضل من أن تضع اتفاقية ترخيص GNU .

3 - **MIT** - Massachusetts Institute of Technology : تتبع هذه الاتفاقية لل GPL و هي تعتبر مكملة لها نذكر أن كل من jQuery - Ruby in Rails و عدد آخر من المشاريع الكبيرة يستخدمون هذه الاتفاقية .

4 - اتفاقية المشاع الابداعي - **Creative Commons (CC)** : هي اتفاقية عامة لا تخص البرامج لكن من الممكن أيضا استعمالها كاطريقة لحفظ الحقوق الفكرية ، يفضل استعمالها أكثر مع المواقع أو الكتب و المؤلفات .

5 – اتفاقية موزيلا العامة - **Mozilla Public License (MPL)** : هي اتفاقية ترخيص تخص شركة موزيلا بمنتاجاتها المختلفة مثل الفايرفوكس و ثاندر بيرد ، يمكنك استخدامها لتنشئ لنفسك اتفاقية ترخيص خاصة بك ، مثل : Yahoo! Public License ، Erlang Public License ، Sun Public License ، فهذه كلها اتفاقيات ترخيص تعتمد على MPL .

6 – اتفاقية ترخيص الأباتشي - Apache Software Foundation (ASF) : هي و احدة من أكثر اتفاقيات الترخيص انتشارا بين البرامج فيوجد الآلاف من البرامج التي تستعملها ، و نذكر أن نظام التشغيل الأندرويد بشهرته الواسعة يستعمل رخصة الأباتشي .

7 – أخيرا اذا أردت أسهل طريقة لترخيص برنامجك فما عليك سوى استعمال هذا السطر الصغير :

Copyright © “Date” “ Company name”, All Rights Reserved.

ما عليك سوى أن تستبدل :
Date ---> بتاريخ المنتج مثل 2015 أو التاريخ منذ إنشائه 1990 – 2015
Company name --> اسم شركتك ، أو أن تضع اسم البرنامج و حسب .

و هكذا تكون قد حفظت حقوق الملكية لبرنامجك .

3 – إطار العمل

تحدثنا في الأجزاء السابقة عن بيئة التطوير المتكاملة (IDE) ، و قلنا أنها عبارة عن برنامج يضم مترجم اللغة و محرر و منقح بالإضافة طبعا لإمكانيات تنظيم ملفات المشروع ... الخ ، أيضا إطار العمل هو بيئة برمجية تساعدك على إنجاز مشروعك بسرعة أكبر لكن إطار العمل موجه لتطبيقات الويب و تصميم المواقع ، عن طريق توفير أكبر قدر ممكن من الأكواد الجاهزة ، طبعا يجب عليك اختيار إطار العمل بعناية لكي يلبي احتياجاتك ، لكن يجب أن أذكر هنا أنك قبل أن تتجه لأطر العمل عليك أولا أن تكون مستوعب تماما للغة و أن تكون على دراية تامة بكيفية إتمام مشروعك بدون إطار العمل ، فإطار العمل دوره مساعدة المبرمج الماهر و ليس اختصار العملية على المبرمج المبتدئ ، طبعا إطار العمل يمكن أن يتضمن مترجم أو مفسر للغة و منقح هذا طبعا بالإضافة لمجموعة من الدوال الجاهزة (API) ، بالنسبة لكيفية برمجة إطار العمل فالعملية هنا تعتمد على "أنماط التصميم - Design pattern " و هي أنظمة ثابتة تستخدم للتخطيط للمشروع و تنظيم كيفية عمل الأكواد ، من أشهرها MVC .

نستعرض معكم الآن مجموعة من أطر العمل في مختلف لغات البرمجة :

1 – C : Saetta Web Server .

2 –C++: Poco , C++CMS .

3 –Java : Spring , Play , AppFuse , Grails .

4 –Python : Django , Pyramid , Tornado .

5 –Perl : Dancer , Mason .

6 –PHP : Laravel , Qcodo , CakePHP , Symfony .

7 –Ruby : Ruby in Rails , Sinatra .

8 – Java Script : Node.js , AngularJS , Express.js .

4 – قواعد البيانات

أحد المواضيع التي لابد أن تقابل المبرمج هي قواعد البيانات ، فالحاسوب كما نعلم هو آلة تستقبل البيانات و تعيدها له بعد عملية المعالجة في هيئة معلومات ، هنا يأتي دور قواعد البيانات فتقريبا لا يخلوا برنامج أو موقع من قاعدة حتى لو كانت صغيرة و غير

معقدة ، و تتنوع قواعد البيانات من حيث الامان ، دعمها للغات و منصات العمل المختلفة ، حجمها و أنواع البيانات التي تدعمها طرق الاتصال بها ، تجد أيضا أن أغلب قواعد البيانات تقسم إلى جداول (table) و الجداول تقسم إلى صفوف (record) و الصفوف تقسم إلى حقول (field) ، لتقريب الفكرة أكثر سنفرض أنه لدينا قاعدة بيانات لمدرسة سنجد أن البيانات تقسم إلى جداول حسب تبعيتها فهناك جدول للطلاب و جدول للمعلمين و جدول للشؤون الدراسية مثلا ، سنجد داخل جدول الطلاب مجموعة من الصفوف حيث أن كل صف يمثل طالب ، و بداخله نجد مجموعة من الحقول التي تمثل البيانات مثل : اسمه ، صفه ، تاريخ ميلاده علاماته الدراسية .. الخ .

إلى هنا أعتقد أن كل شيء واضح و جميل ، لكن كان هناك مشكلة و هي أن لغات البرمجة لا تستطيع أن تتصل مع قواعد البيانات بسهولة ، فالأمر كان طويل و ممل ، لذا طور العلماء كالعادة☺ حل عبقري ألا و هو محرك قواعد البيانات و كانت هذه المحركات لديها القدرة على فعل كل ما نريد من مهام كإضافة و حذف و تعديل الجداول و الحقول .

من كل ما سبق نجد أنه تكون لدينا نظام لغدارة قواعد البيانات (DataBase Mangement System) هذه الانظمة لديها دوال خاصة و مكتبات مساعدة تمكن المبرمج من التعامل بكل يسر و سهولة من بياناته داخل القاعدة ن من أشهر هذه الأنظمة :

- **Microsoft Access**
- **Microsoft SQL Server**
- **MySQL**
- **Oracle**
- **SQLite**
- **Firebird**

5 – واجهة برمجة التطبيق

أو كما نعرفها (API – Application programming interface) ، هي مجموعة من الدوال المعدة لتسهيل مجموعة من العمليات على المبرمج مثل : جعل البرنامج يعمل مع بدأ تشغيل الحاسب ، تمكين البرنامج من الوصول لإعدادات معينة داخل نظام التشغيل و غيره الكثير من الامور التي يصعب على المبرمج القيام بها أو لنقل أنها ستأخذ الكثير من العمل و الجهد .

هذه الواجهة أو لنقل الدوال ليست فقط مرتبطة بنظام تشغيل ، بل أي برنامج أو موقع أو حتى لعبة تستطيع بناء دوالAPI خاصة بها كي تمكن المبرمجين من استعمال أجزاء أو لنقل خدمات من التطبيق في تطبيقاتهم .

من أكثر دوال API انتشارا ، تلك المتعلقة بمواقع YouTube , Google Map , FaceBook , Yahoo حيث تمكنك هذه المواقع من الحصول على بعض الخدمات يمكن اضافتها في تطبيقاتك مثل معرفة أحوال الطقس أو تحديد موقع ما ، لدينا أيضا العملاق windows فهو يدعم الكثير من الدوال و التي تظهر فائدتها كثيرا عند استعمالك لبيئة التطوير Visual studio فتجد مثلا في C++ أنك تستطيع بناء برنامج كامل فقط بدوال API .

حسنا بالنسبة لكيفية بناء هذه الدوال ☺ يعني كيف اصنع لتطبيقي دوال API ؟؟؟

عملية بناء الدوال غير محددة ، فكل مبرمج يستطيع عمل أفكار خاصة به لكن بشكل عام الفكرة تتمحور حول جعل أجزاء من كود البرنامج قادرة على تصدير المخرجات لتطبيق آخر ، غالبا ما تستعمل ملفات بصيغة XML لتصدير المدخلات و استقبال المخرجات مثلا أنت تريد حالة الطقس في بلد ما كل ما تفعله هو أن تكتب الدالة و بداخلها مثلا اسم البلد ، الدالة تكون مرتبطة بخادم (server) يقوم باستقبال الاستعلام المرسل إليه و بناء على اسم البلد سوف يجلب أحوال الطقس و يخزنها في ملف و يتم إرسالها إلى تطبيقك و في النهاية تستخرج أنت هذه البيانات و تقدمها للمستخدم .

هذه هي الفكرة بكل بساطة أنك تتيح كودك للآخرين قد تجد بعض هذه الدوال تحتاج لاشتراك مسبق تدفع له مبلغ من المال و قد تكون بحاجة لامتلاك حساب دون الحاجة لدفع المال مثل Google API .

AGASHE

أيضا قد يمر عليك مصطلح (شبكات تواصل المحتوى - CDN - **content delivery network**) ، **هنا الفكرة تختلف قليلا فنحن** هنا نستخدم هذه البيانات داخل الكود و ليس فقط داخل التطبيق يعني على عكس API التي تؤدي مهمة و ينتهي دورها هنا الأمر مختلف فهذه البيانات في الغالب عبارة عن مكتبات برمجية يستعملها المبرمج أثناء البرنامج ، فبدل أن يعيد تحميل المكتبة كل مرة لكل مستخدم ، فهو يقوم باستدعائها مرة و احدة و هذا يقلل من العبء على التطبيق ، ستجد أن هذه الفكرة منتشرة كثيرا بين المواقع خاصة بالنسبة لكل من Bootstrap و jquery ، و قليل ما تراها لتطبيقات سطح المكتب .

بالعودة لكيفية عمل دوال API في حالة البرامج فالموضوع سهل نسبيا لان كل شيء على جهاز واحد ، لكن ماذا عن تطبيقات الويب حيث أن دوالك التي برمجتها موجودة في أجهزة (server) و مواقع المستخدمين في أجهزة (server) أخرى ، نجد هنا الحل في استعمال قواعد إرسال البيانات و نذكر من أشهرها (**SOAP** - *Simple Object Access protocol*) هو عبارة عن مجموعة من القواعد التي تستخدمها في تنظيم كيفية إرسال و استقبال المعلومات لتوفير الحماية لموقعك و مواقع المستخدمين ، كما انه يوفر الكثير من الطرق لارسال و استقبال البيانات ليس فقط ملفات XML فهناك HTTP Requests و Simple Mail Transfer Protocol (SMTP), .

إلى هنا ينتهي هذا الجزء ، أرجوا من الله أن تكونوا قد استفدتوا من هذه المعلومات و أن تواصلوا البحث أكثر فالبرمجة بحر واسع لا نهاية له ، هذا و أتمنى التوفيق لي و لكم و السلام عليكم و رحمة الله تعالى و بركاته .

ملحوظة : جميع الصور في هذا الكتاب هي من مصادر خارجية و حقوقها محفوظة لناشريها

للتواصل معي و في حالة وجود أي استفسار أو سؤال :

Modi401@hotmail.com

أو

https://www.facebook.com/mohamed.yossef.583

AGASHE

2010 – 2015