Take Care with Char

ESC101: Fundamentals of Computing

Purushottam Kar

Mid-sem Theory Exam

- September 22nd, 2018 (Saturday)
- Time: 1PM – 3PM
- Room: announced shortly
- Syllabus: till whatever is covered till Sep 14th tutorial
- No Make-up Exam – do not miss this exam
- Open handwritten notes – no printouts, mobiles, iPads.

Doubt-clearing Session

- Date: September 15th, 2018 (coming Saturday)
- Time: 5PM – 7PM
- Room: CC-02
- Students not comfortable with English are welcome
- Other students also welcome to clear doubts
- Please revise and have list of doubts before coming
- Will not cover lectures again in detail – only doubts

Advanced Track

- 29 students selected for advanced track
- Sorry for delay in sending out mentor allocation
- Had a marathon 4 hours grading session last night ☺
- Will definitely send out notifications this afternoon

Char: new datatype

Char: new datatype

Close cousin of the int and long datatypes

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

```
#include <stdio.h>
```

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

```
#include <stdio.h>
int main(){
```

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

```
#include <stdio.h>
int main(){
char a = 'p';
```

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

```
#include <stdio.h>
int main(){
char a = ‘p’;
scanf(“%c”, &a);
```

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

```
#include <stdio.h>
int main(){
char a = ‘p’;
scanf(“%c”, &a);
printf(“My first char %c”, a);
```

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

```
#include <stdio.h>
int main(){
char a = ‘p’;
scanf(“%c”, &a);
printf(“My first char %c”, a);
return 0;
```

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

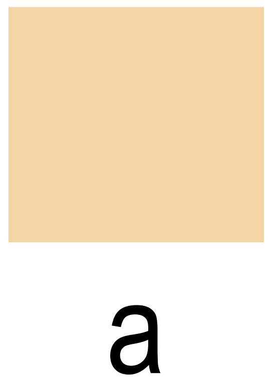
```
#include <stdio.h>
int main(){
char a = ‘p’;
scanf(“%c”, &a);
printf(“My first char %c”, a);
return 0;
}
```

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

```
#include <stdio.h>
int main(){
char a = ‘p’;
scanf(“%c”, &a);
printf(“My first char %c”, a);
return 0;
}
```

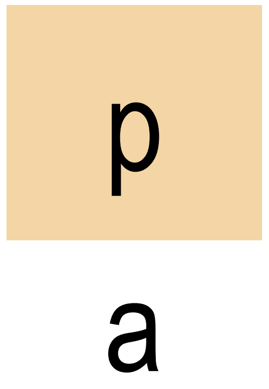


Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

```
#include <stdio.h>
int main(){
char a = ‘p’;
scanf(“%c”, &a);
printf(“My first char %c”, a);
return 0;
}
```

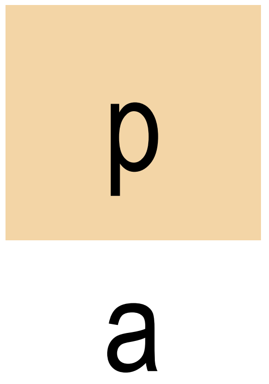


Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

```
#include <stdio.h>
int main(){
char a = ‘p’;
scanf(“%c”, &a);
printf(“My first char %c”, a);
return 0;
}
```



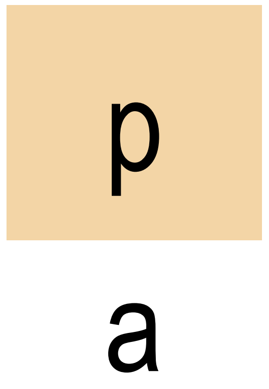
%c

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

```
#include <stdio.h>
int main(){
char a = ‘p’;
scanf(“%c”, &a);
printf(“My first char %c”, a);
return 0;
}
```



%c

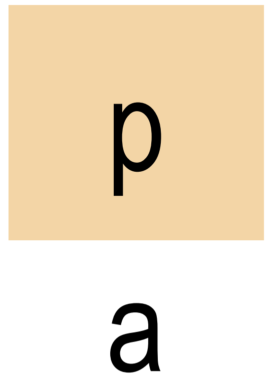
Char constants enclosed in ‘ ’

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

```
#include <stdio.h>
int main(){
char a = ‘p’;
scanf(“%c”, &a);
printf(“My first char %c”, a);
return 0;
}
```



%c

Char constants enclosed in ‘ ’

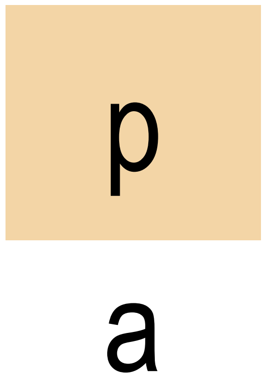
Integer arithmetic applies to char as well +, -, /, *, %, ()

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

```
#include <stdio.h>
int main(){
char a = ‘p’;
scanf(“%c”, &a);
printf(“My first char %c”, a);
return 0;
}
```



%c

Char constants enclosed in ‘ ’

Integer arithmetic applies to char as well +, -, /, *, %, ()

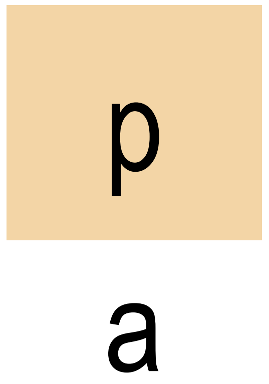
Can use it for nice tricks but be careful

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

```
#include <stdio.h>
int main(){
char a = ‘p’;
scanf(“%c”, &a);
printf(“My first char %c”, a);
return 0;
}
```



%c

Char constants enclosed in ‘ ’

Integer arithmetic applies to char as well +, -, /, *, %, ()

Can use it for nice tricks but be careful

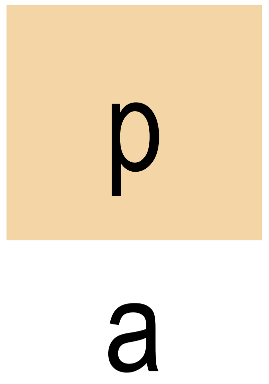
Can typecast to/from char

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

```
#include <stdio.h>
int main(){
char a = ‘p’;
scanf(“%c”, &a);
printf(“My first char %c”, a);
return 0;
}
```



%c

Char constants enclosed in ‘ ’

Integer arithmetic applies to char as well +, -, /, *, %, ()

Can use it for nice tricks but be careful

Can typecast to/from char

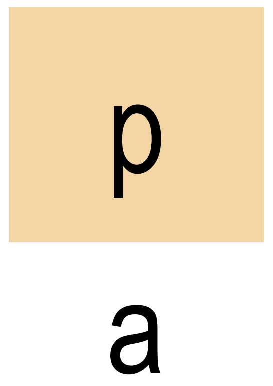
Can have char arrays – strings

Char: new datatype

Close cousin of the int and long datatypes

Internally stored as an integer between 0 and 127

```
#include <stdio.h>
int main(){
char a = ‘p’;
scanf(“%c”, &a);
printf(“My first char %c”, a);
return 0;
}
```



%c

Char constants enclosed in ‘ ’

Integer arithmetic applies to char as well +, -, /, *, %, ()

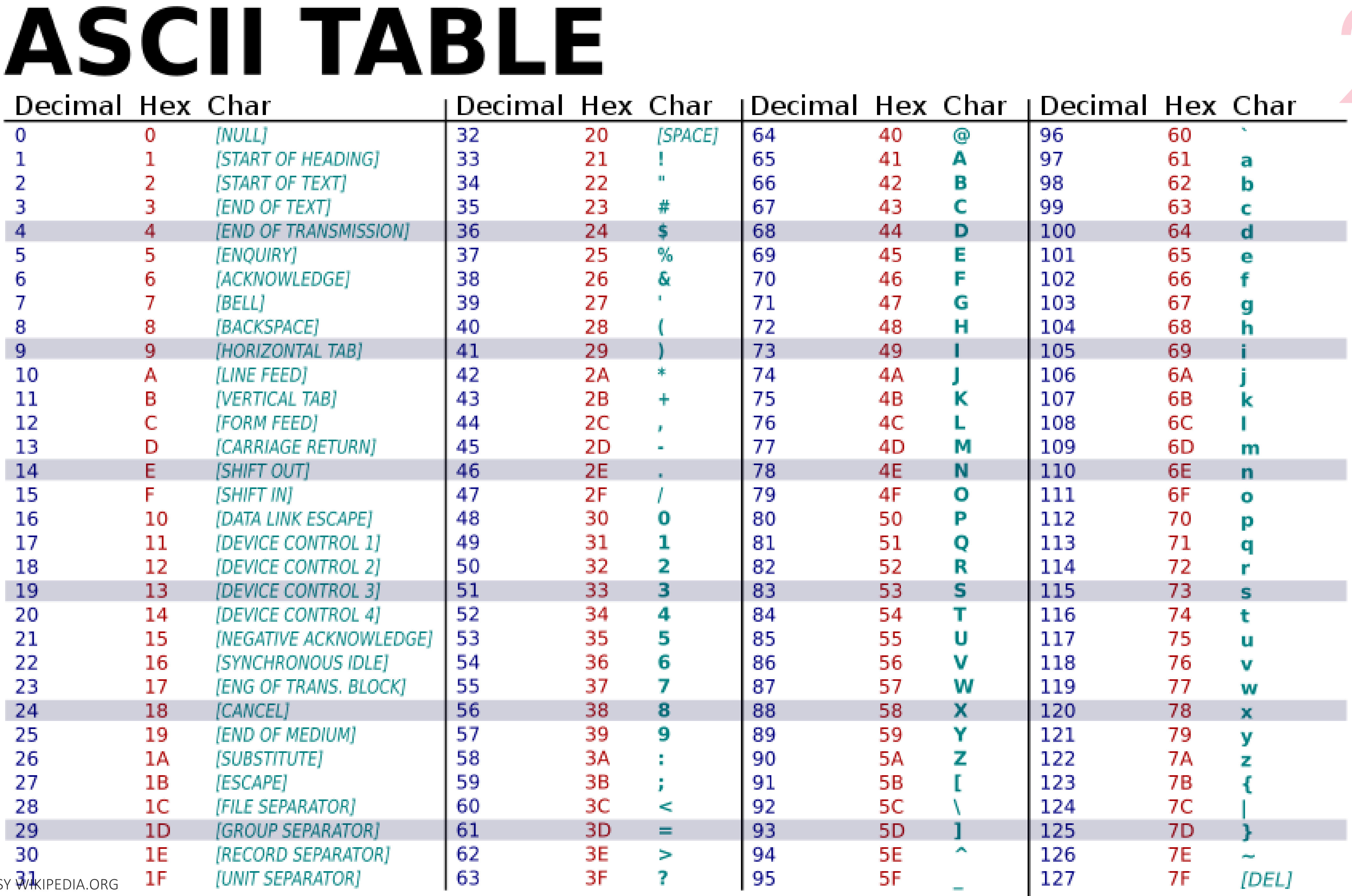
Can use it for nice tricks but be careful

Can typecast to/from char

Can have char arrays – strings

Case sensitive ‘a’, ‘A’ different

IMAGE COURTESY WIKIPEDIA.ORG



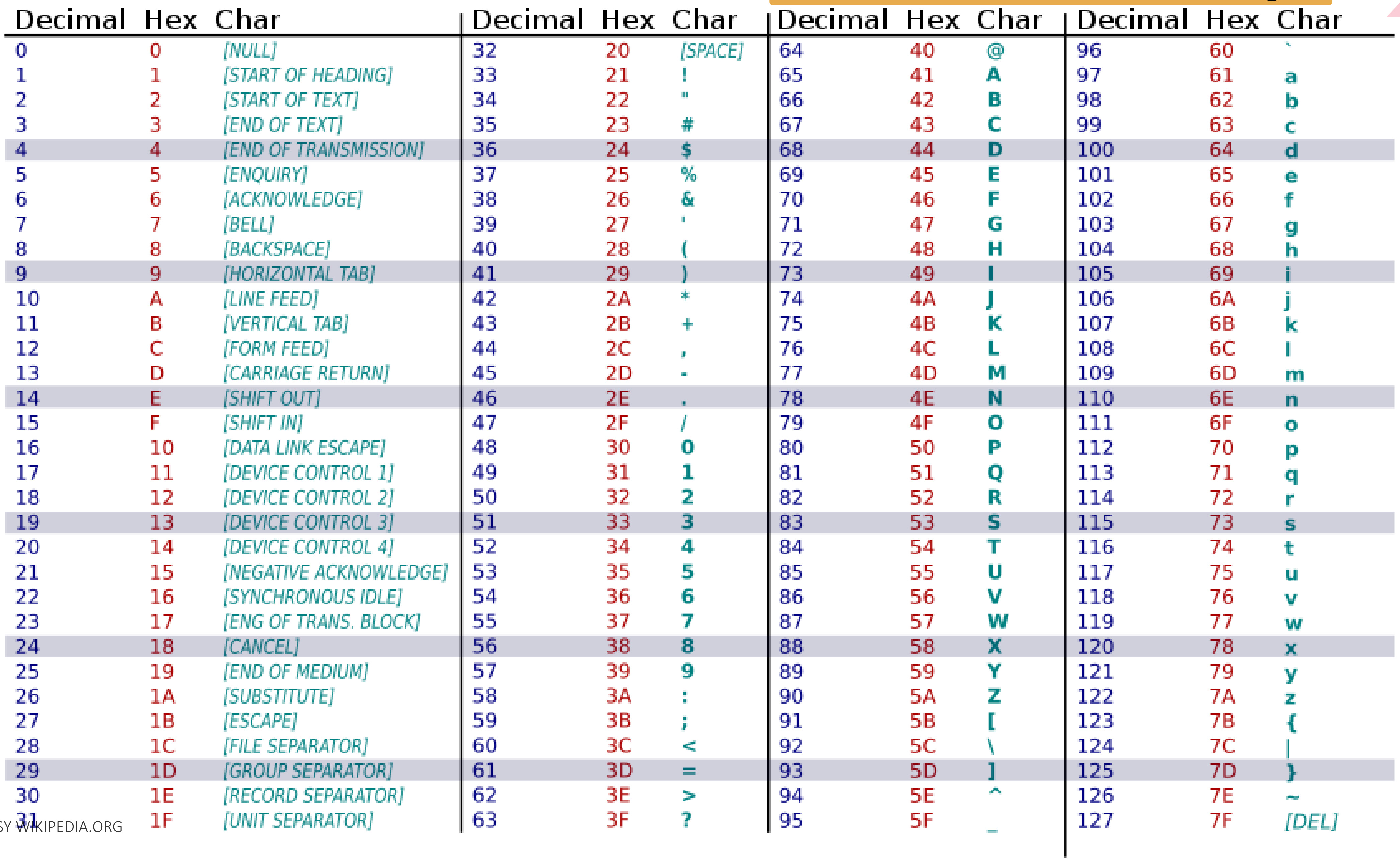
ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

IMAGE COURTESY WIKIPEDIA.ORG

ASCII TABLE

American Standard Code for Information Interchange

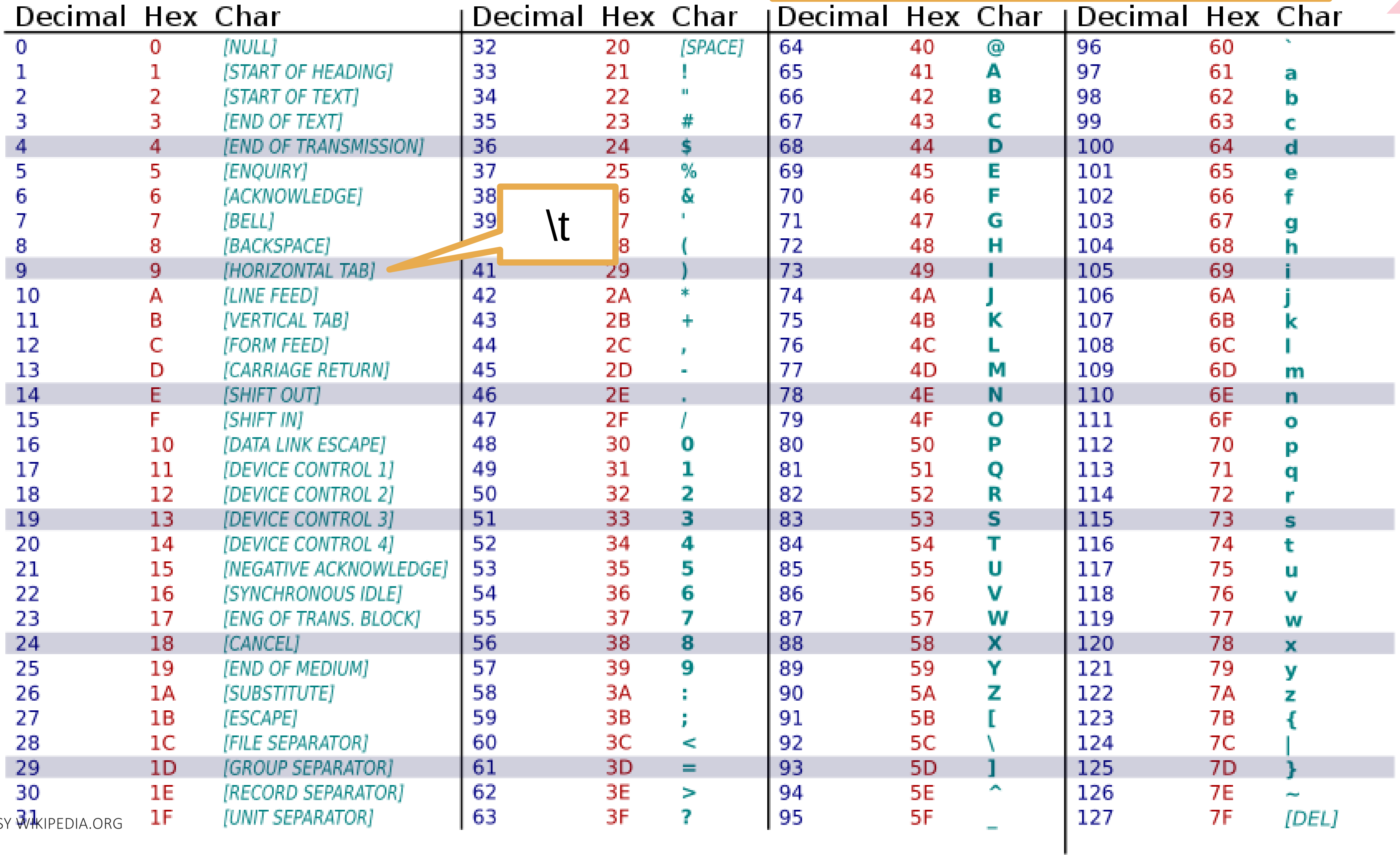


Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

IMAGE COURTESY WIKIPEDIA.ORG

ASCII TABLE

American Standard Code for Information Interchange



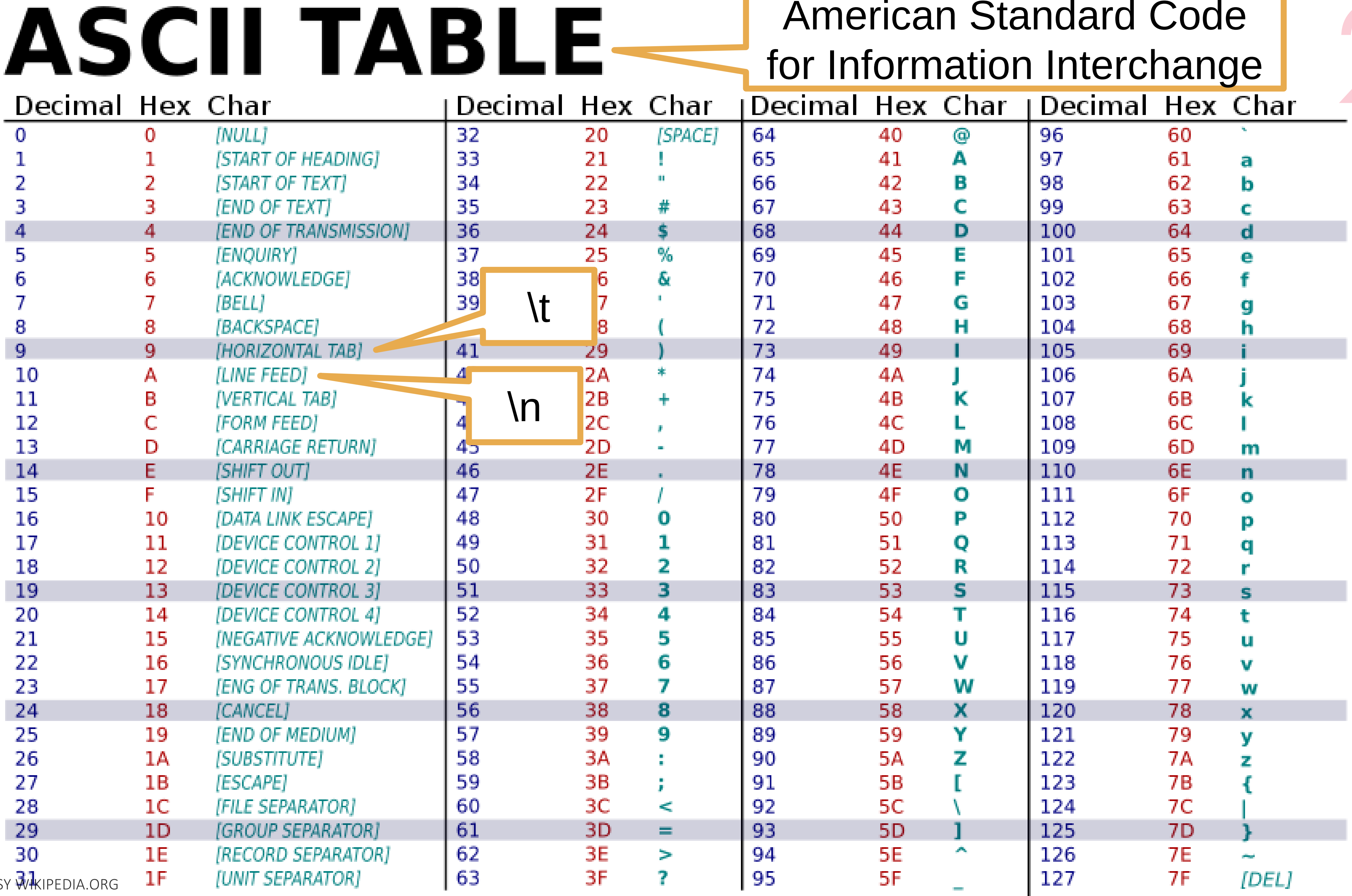
Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	6	&	70	46	F	102	66	f
7	7	[BELL]	39	7	'	71	47	G	103	67	g
8	8	[BACKSPACE]	[illegible]	8	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

\t

IMAGE COURTESY WIKIPEDIA.ORG

ASCII TABLE

American Standard Code for Information Interchange



Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	6	&	70	46	F	102	66	f
7	7	[BELL]	39	7	'	71	47	G	103	67	g
8	8	[BACKSPACE]		8	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	4	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	4	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	4	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

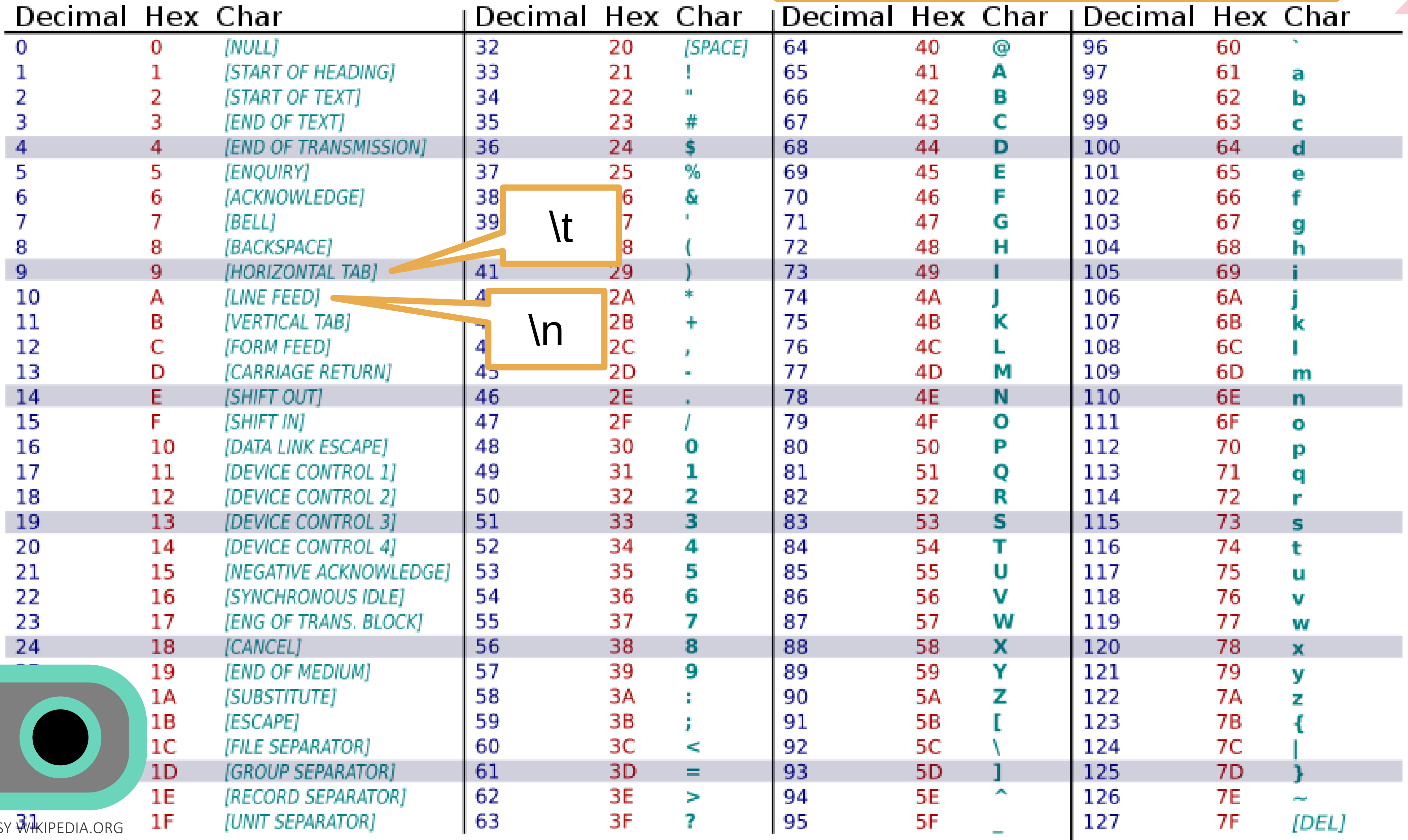
\t

\n

IMAGE COURTESY WIKIPEDIA.ORG

ASCII TABLE

American Standard Code for Information Interchange



Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	6	&	70	46	F	102	66	f
7	7	[BELL]	39	7	'	71	47	G	103	67	g
8	8	[BACKSPACE]	[illegible]	8	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	4	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	[illegible]	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	4	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
[illegible]	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
[illegible]	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
[illegible]	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
[illegible]	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
[illegible]	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
[illegible]	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

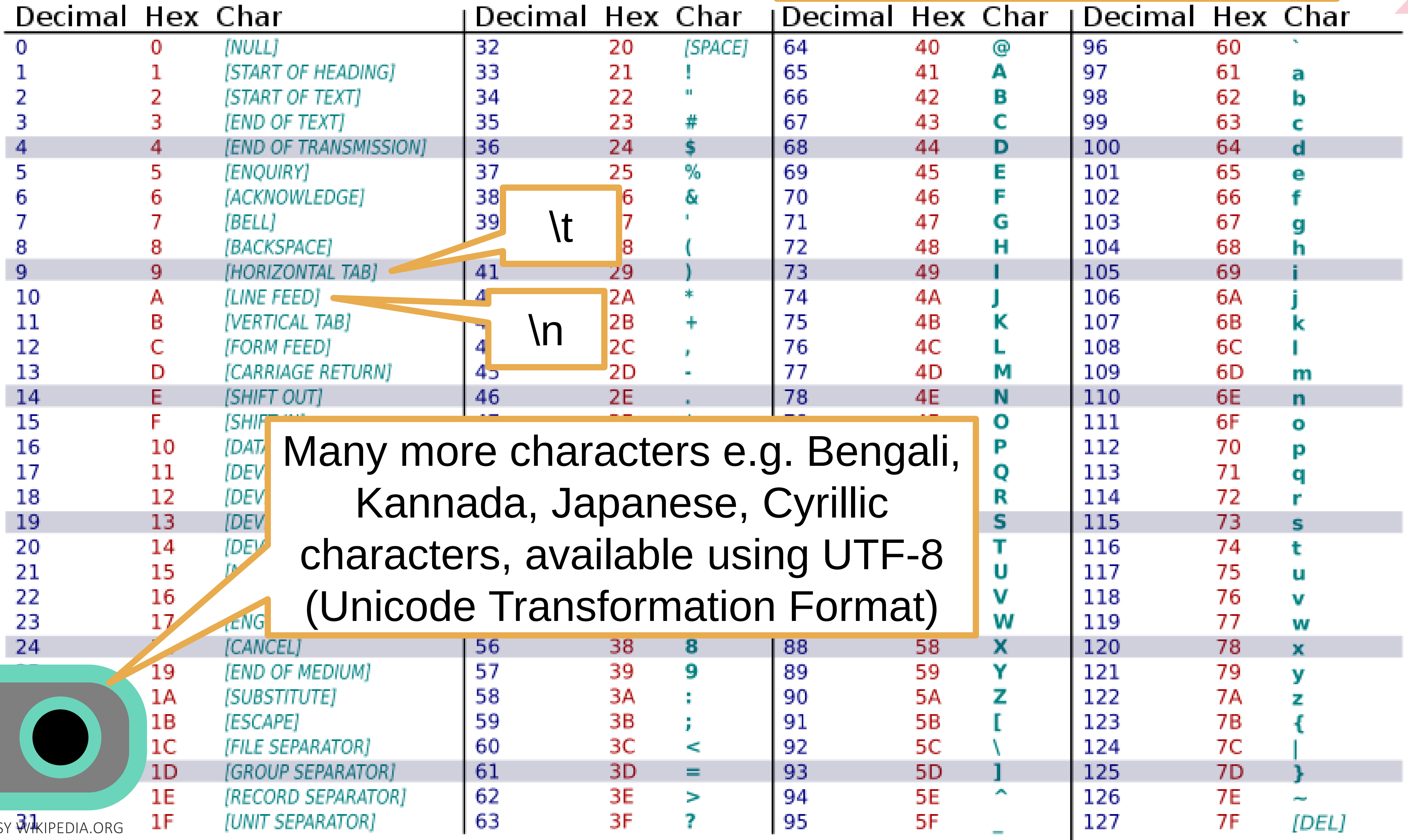
\t

\n

IMAGE COURTESY WIKIPEDIA.ORG

ASCII TABLE

American Standard Code for Information Interchange



Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	[illegible]	&	70	46	F	102	66	f
7	7	[BELL]	39	[illegible]	'	71	47	G	103	67	g
8	8	[BACKSPACE]	[illegible]	[illegible]	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	[illegible]	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	[illegible]	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	[illegible]	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	[illegible]	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	O	111	6F	o
16	10	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	P	112	70	p
17	11	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Q	113	71	q
18	12	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	R	114	72	r
19	13	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	S	115	73	s
20	14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	T	116	74	t
21	15	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	U	117	75	u
22	16	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	V	118	76	v
23	17	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	W	119	77	w
24	[illegible]	[CANCEL]	56	38	8	88	58	X	120	78	x
[illegible]	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
[illegible]	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
[illegible]	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
[illegible]	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
[illegible]	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
[illegible]	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

\t

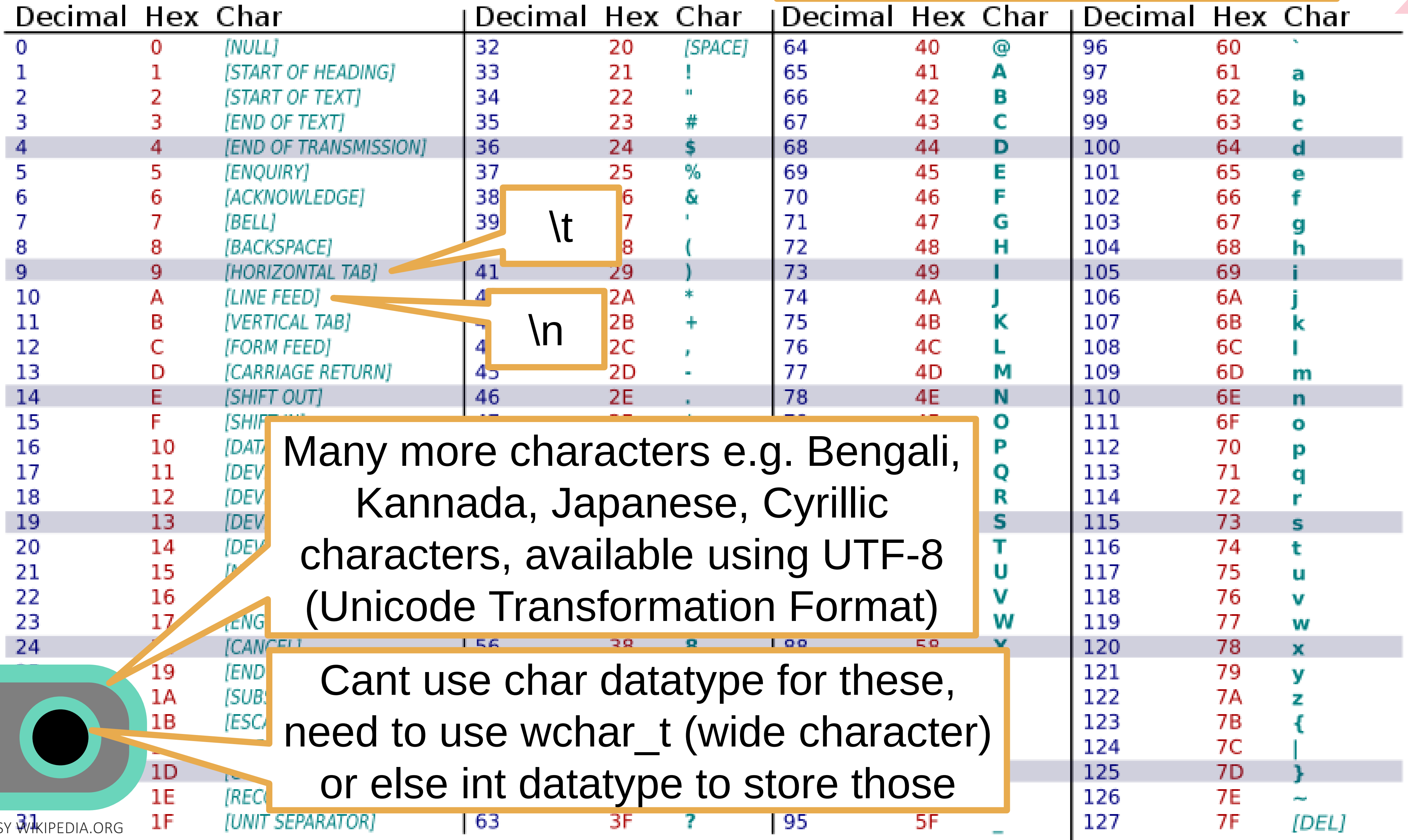
\n

Many more characters e.g. Bengali, Kannada, Japanese, Cyrillic characters, available using UTF-8 (Unicode Transformation Format)

IMAGE COURTESY WIKIPEDIA.ORG

ASCII TABLE

American Standard Code for Information Interchange

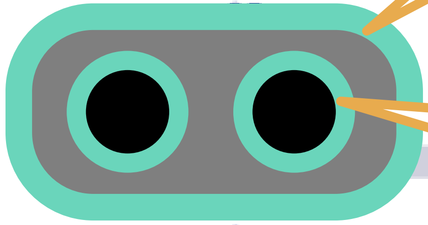


Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	6	&	70	46	F	102	66	f
7	7	[BELL]	39	7	'	71	47	G	103	67	g
8	8	[BACKSPACE]	[illegible]	8	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	4	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	[illegible]	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	4	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIF	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	O	111	6F	o
16	10	[DAT	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	P	112	70	p
17	11	[DEV	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Q	113	71	q
18	12	[DEV	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	R	114	72	r
19	13	[DEV	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	S	115	73	s
20	14	[DEV	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	T	116	74	t
21	15	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	U	117	75	u
22	16	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	V	118	76	v
23	17	[ENG	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	W	119	77	w
24	[illegible]	[CANCEL]	56	38	8	88	58	X	120	78	x
[illegible]	19	[END	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	121	79	y
[illegible]	1A	[SUB	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	122	7A	z
[illegible]	1B	[ESC	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	123	7B	{
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	124	7C	|
[illegible]	1D	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	125	7D	}
[illegible]	1E	[REC	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

\t

\n

Many more characters e.g. Bengali, Kannada, Japanese, Cyrillic characters, available using UTF-8 (Unicode Transformation Format)



Cant use char datatype for these, need to use wchar_t (wide character) or else int datatype to store those

IMAGE COURTESY WIKIPEDIA.ORG

Take care with char

Take care with char

char a = p; Mr C will search for a variable named p.

Take care with char

char a = p; Mr C will search for a variable named p.

To assign character constant ‘p’ to a, char a = ‘p’;

Take care with char

`char a = p;` Mr C will search for a variable named p.

To assign character constant ‘p’ to a, `char a = ‘p’;`

Note that ‘5’ and 5 are different according to Mr C

Take care with char

char a = p; Mr C will search for a variable named p.

To assign character constant ‘p’ to a, char a = ‘p’;

Note that ‘5’ and 5 are different according to Mr C

- ‘5’ is a character constant stored internally as the integer 53

Take care with char

char a = p; Mr C will search for a variable named p.

To assign character constant ‘p’ to a, char a = ‘p’;

Note that ‘5’ and 5 are different according to Mr C

- ‘5’ is a character constant stored internally as the integer 53
- 5 is an integer constant stored internally as the integer 5 itself

Take care with char

char a = p; Mr C will search for a variable named p.

To assign character constant ‘p’ to a, char a = ‘p’;

Note that ‘5’ and 5 are different according to Mr C

- ‘5’ is a character constant stored internally as the integer 53
- 5 is an integer constant stored internally as the integer 5 itself

Be very careful if mixing %c with other format specifiers like %d, %ld, %f, %lf in scanf and printf statements

Take care with char

char a = p; Mr C will search for a variable named p.

To assign character constant ‘p’ to a, char a = ‘p’;

Note that ‘5’ and 5 are different according to Mr C

- ‘5’ is a character constant stored internally as the integer 53
- 5 is an integer constant stored internally as the integer 5 itself

Be very careful if mixing %c with other format specifiers like %d, %ld, %f, %lf in scanf and printf statements

getchar(), putchar() shortcuts to read/print single char

Take care with char

char a = p; Mr C will search for a variable named p.

To assign character constant ‘p’ to a, char a = ‘p’;

Note that ‘5’ and 5 are different according to Mr C

- ‘5’ is a character constant stored internally as the integer 53
- 5 is an integer constant stored internally as the integer 5 itself

Be very careful if mixing %c with other format specifiers like %d, %ld, %f, %lf in scanf and printf statements

getchar(), putchar() shortcuts to read/print single char

When using characters in arithmetic, relational, logical expressions, integer (ASCII) value of character gets used

Take care with char

Take care with char

Tip on using char: if output wrong with %c, try printing the same char with %d instead. Its decimal ASCII value will get printed which may help you find error

Take care with char

Tip on using char: if output wrong with %c, try printing the same char with %d instead. Its decimal ASCII value will get printed which may help you find error

Since char is stored internally as integer

Take care with char

Tip on using char: if output wrong with %c, try printing the same char with %d instead. Its decimal ASCII value will get printed which may help you find error

Since char is stored internally as integer

```
char abc = 'p';
printf("%d", abc);
```

Take care with char

Tip on using char: if output wrong with %c, try printing the same char with %d instead. Its decimal ASCII value will get printed which may help you find error

Since char is stored internally as integer

```
char abc = 'p';
printf("%d", abc);
```

will print the ASCII value of the character stored in abc

Take care with char

Tip on using char: if output wrong with %c, try printing the same char with %d instead. Its decimal ASCII value will get printed which may help you find error

Since char is stored internally as integer

```
char abc = 'p';
printf("%d", abc);
```

will print the ASCII value of the character stored in abc

```
printf("%c",65)
```

Take care with char

Tip on using char: if output wrong with %c, try printing the same char with %d instead. Its decimal ASCII value will get printed which may help you find error

Since char is stored internally as integer

char abc = ‘p’;

printf("%d", abc);

will print the ASCII value of the character stored in abc

printf("%c",65) will print character A (implicit typecasting)

Take care with char

Tip on using char: if output wrong with %c, try printing the same char with %d instead. Its decimal ASCII value will get printed which may help you find error

Since char is stored internally as integer

```
char abc = ‘p’;
printf("%d", abc);
```

will print the ASCII value of the character stored in abc

`printf("%c",65)` will print character A (implicit typecasting)

Warning: Implicit typecasting may not always work

Take care with char

Tip on using char: if output wrong with %c, try printing the same char with %d instead. Its decimal ASCII value will get printed which may help you find error

Since char is stored internally as integer

```
char abc = ‘p’;
printf("%d", abc);
```

will print the ASCII value of the character stored in abc

`printf("%c",65)` will print character A (implicit typecasting)

Warning: Implicit typecasting may not always work

Exercise: write code to flip the case i.e. A→a, d→D etc

Character Arrays

Character Arrays

All things we learnt about int/float arrays apply here too

Character Arrays

All things we learnt about int/float arrays apply here too

However, much more exciting things can be done here

Character Arrays

All things we learnt about int/float arrays apply here too

However, much more exciting things can be done here

Char arrays also called *strings (well … almost all of them)*

Character Arrays

All things we learnt about int/float arrays apply here too

However, much more exciting things can be done here

Char arrays also called *strings (well … almost all of them)*

English word *string* means a thread or a collection of items put together. *The pearls were strung together.*

Character Arrays

All things we learnt about int/float arrays apply here too

However, much more exciting things can be done here

Char arrays also called *strings (well … almost all of them)*

English word *string* means a thread or a collection of items put together. *The pearls were strung together.*

In C, string implies a character array (well … almost)

Character Arrays

All things we learnt about int/float arrays apply here too

However, much more exciting things can be done here

Char arrays also called *strings (well … almost all of them)*

English word *string* means a thread or a collection of items put together. *The pearls were strung together.*

In C, string implies a character array (well … almost)

Note: string is **not a datatype** in C

Character Arrays

All things we learnt about int/float arrays apply here too

However, much more exciting things can be done here

Char arrays also called *strings (well … almost all of them)*

English word *string* means a thread or a collection of items put together. *The pearls were strung together.*

In C, string implies a character array (well … almost)

Note: string is **not a datatype** in C

Word string is **not a keyword** in C

Character Arrays

All things we learnt about int/float arrays apply here too

However, much more exciting things can be done here

Char arrays also called *strings (well … almost all of them)*

English word *string* means a thread or a collection of items put together. *The pearls were strung together.*

In C, string implies a character array (well … almost)

Note: string is **not a datatype** in C

Word string is **not a keyword** in C

```
int string = 0;
```

Character Arrays

All things we learnt about int/float arrays apply here too

However, much more exciting things can be done here

Char arrays also called *strings (well … almost all of them)*

English word *string* means a thread or a collection of items put together. *The pearls were strung together.*

In C, string implies a character array (well … almost)

Note: string is **not a datatype** in C

Word string is **not a keyword** in C

```
int string = 0;
```
✓

Declaring and Using Strings

Declaring and Using Strings

Can be initialized at time of declaration

Declaring and Using Strings

Can be initialized at time of declaration

```
char str[50] = {‘H’,’e’,’l’,’l’,’o’,’ ‘,‘W’,’o’,’r’,’l’,’d’};
```

Declaring and Using Strings

Can be initialized at time of declaration

```
char str[50] = {‘H’,’e’,’l’,’l’,’o’,’ ‘,‘W’,’o’,’r’,’l’,’d’};
char str[50] = "Hello World";
```

Declaring and Using Strings

Can be initialized at time of declaration

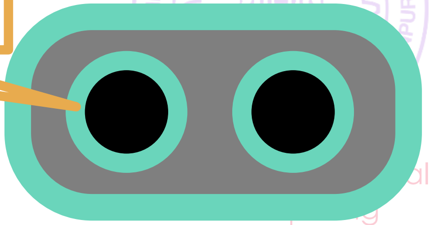
```
char str[50] = {‘H’,’e’,’l’,’l’,’o’,’ ‘,‘W’,’o’,’r’,’l’,’d’};
char str[50] = "Hello World";
```

Declaring and Using Strings

Can be initialized at time of declaration

```
char str[50] = {‘H’,’e’,’l’,’l’,’o’,’ ‘,‘W’,’o’,’r’,’l’,’d’};
char str[50] = "Hello World";
```

Partly initialized since only 11 characters this phrase

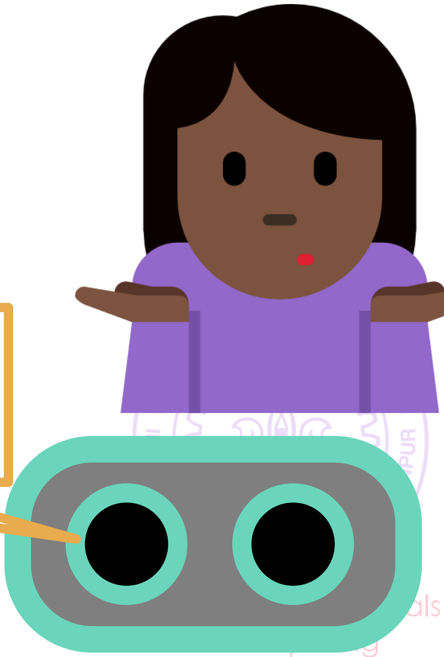


Declaring and Using Strings

Can be initialized at time of declaration

```
char str[50] = {‘H’,’e’,’l’,’l’,’o’,’ ‘,‘W’,’o’,’r’,’l’,’d’};
char str[50] = "Hello World";
```

Partly initialized since only 11 characters this phrase



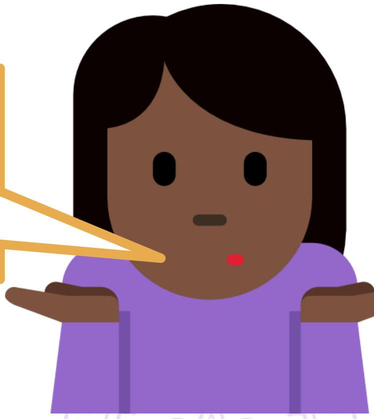
Declaring and Using Strings

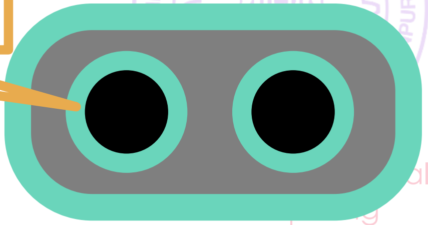
Can be initialized at time of declaration

```
char str[50] = {‘H’,’e’,’l’,’l’,’o’,’ ‘,‘W’,’o’,’r’,’l’,’d’};
char str[50] = "Hello World";
```

Hello has 5 characters, World has 5, what is the 11th character?

Partly initialized since only 11 characters this phrase

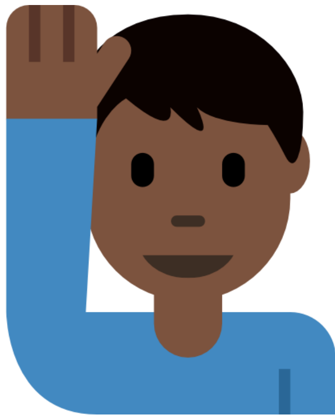




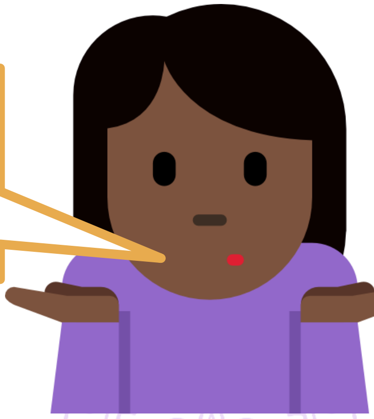
Declaring and Using Strings

Can be initialized at time of declaration

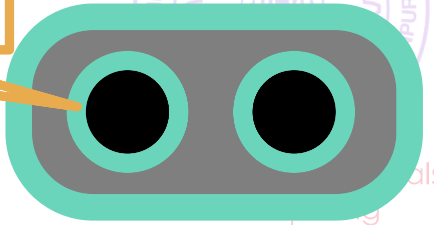
```
char str[50] = {‘H’,’e’,’l’,’l’,’o’,’ ‘,‘W’,’o’,’r’,’l’,’d’};
char str[50] = "Hello World";
```



Hello has 5 characters, World has 5, what is the 11th character?



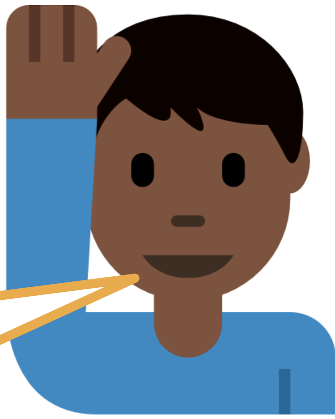
Partly initialized since only 11 characters this phrase



Declaring and Using Strings

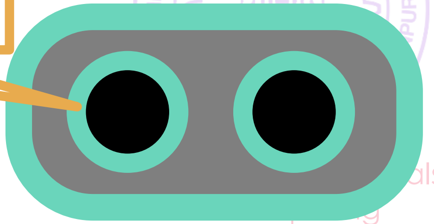
Can be initialized at time of declaration

```
char str[50] = {‘H’,’e’,’l’,’l’,’o’,’ ‘,‘W’,’o’,’r’,’l’,’d’};
char str[50] = "Hello World";
```



The space between the two words. Space is a character too!

Hello has 5 characters, World has 5, what is the 11th character?

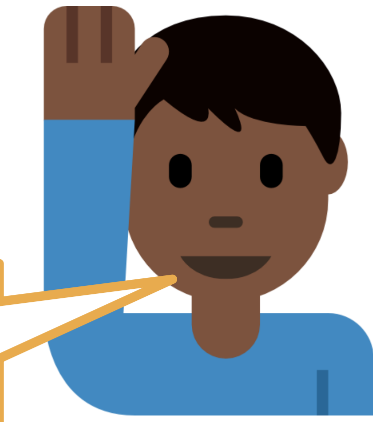


Partly initialized since only 11 characters this phrase

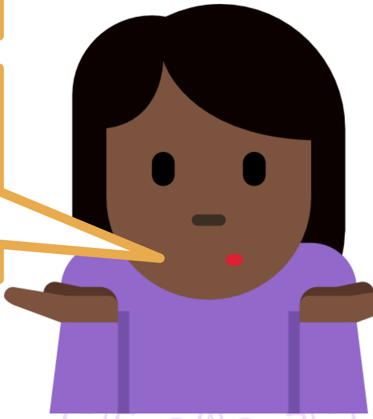
Declaring and Using Strings

Can be initialized at time of declaration

```
char str[50] = {‘H’,’e’,’l’,’l’,’o’,’ ‘,‘W’,’o’,’r’,’l’,’d’};
char str[50] = "Hello World";
```

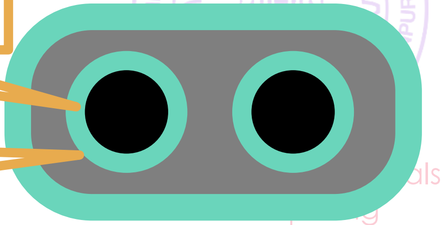


The space between the two words. Space is a character too!



Hello has 5 characters, World has 5, what is the 11th character?

Partly initialized since only 11 characters this phrase



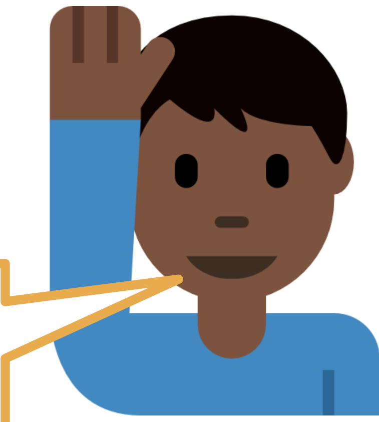
Very good!

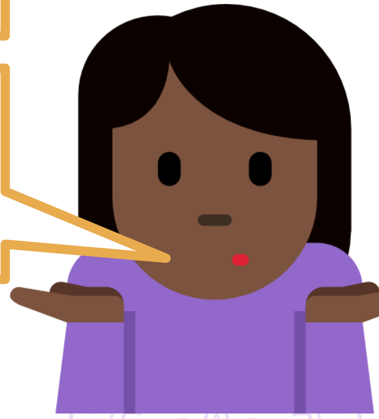
Declaring and Using Strings

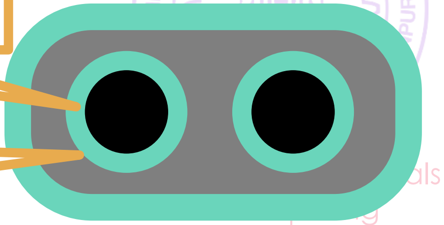
Can be initialized at time of declaration

```
char str[50] = {‘H’,’e’,’l’,’l’,’o’,’ ‘,‘W’,’o’,’r’,’l’,’d’};
char str[50] = "Hello World";
```

Just like int arrays, char arrays cannot be initialized this way after declaration is done




Hello has 5 characters, World has 5, what is the 11th character?



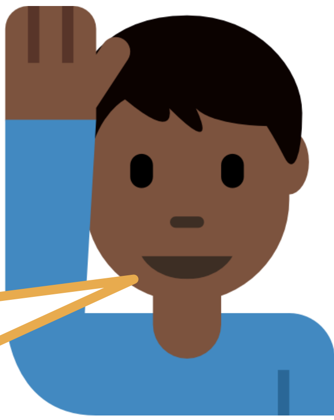
Declaring and Using Strings

Can be initialized at time of declaration

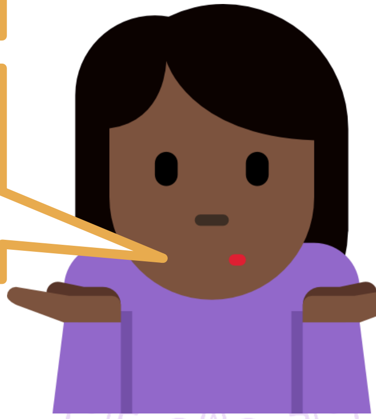
```
char str[50] = {‘H’,’e’,’l’,’l’,’o’,’ ‘,‘W’,’o’,’r’,’l’,’d’};
char str[50] = "Hello World";
```

Just like int arrays, char arrays cannot be initialized this way after declaration is done

Other ways: scanf (with %s), gets

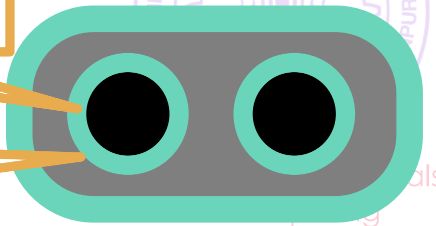


The space between the two words. Space is a character too!



Hello has 5 characters, World has 5, what is the 11th character?

Partly initialized since only 11 characters this phrase



Very good!

Declaring and Using Strings

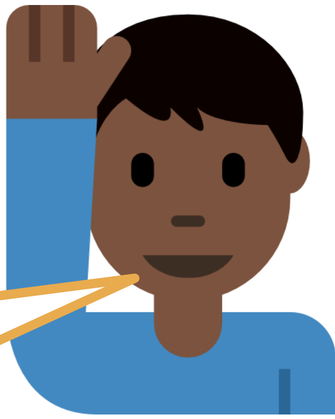
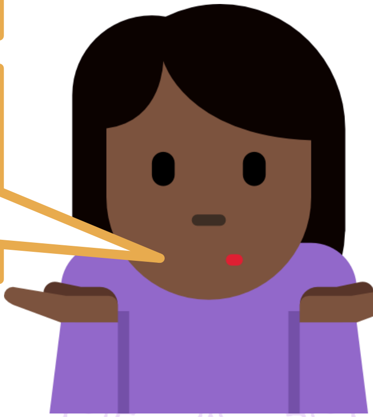
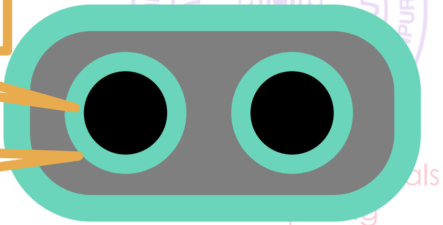
Can be initialized at time of declaration

```
char str[50] = {‘H’,’e’,’l’,’l’,’o’,’ ‘,‘W’,’o’,’r’,’l’,’d’};
char str[50] = "Hello World";
```

Just like int arrays, char arrays cannot be initialized this way after declaration is done

Other ways: scanf (with %s), gets

Both are very unsafe – crash!

The space between the two words. Space is a character too!

Hello has 5 characters, World has 5, what is the 11th character?

Partly initialized since only 11 characters this phrase

Very good!

Declaring and Using Strings

Can be initialized at time of declaration

```
char str[50] = {'H','e','l','l','o',' ','W','o','r','l','d'};
char str[50] = "Hello World";
```

Just like int arrays, char arrays cannot be initialized this way after declaration is done

Other ways: scanf (with %s), gets

Both are very unsafe – crash!

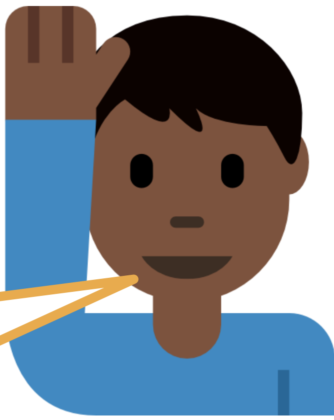
To print: puts, printf (with %s)

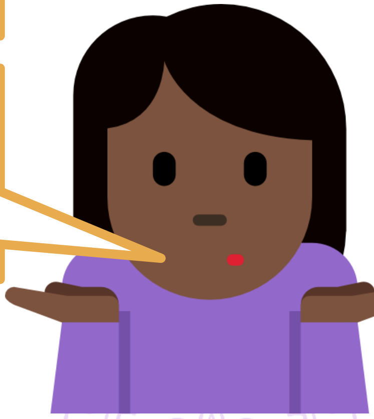
The space between the two words. Space is a character too!

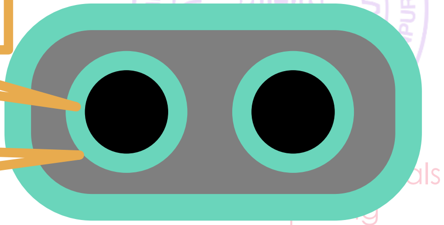
Hello has 5 characters, World has 5, what is the 11th character?

Partly initialized since only 11 characters this phrase

Very good!







Declaring and Using Strings

Can be initialized at time of declaration

```
char str[50] = {‘H’,’e’,’l’,’l’,’o’,’ ‘,‘W’,’o’,’r’,’l’,’d’};
char str[50] = "Hello World";
```

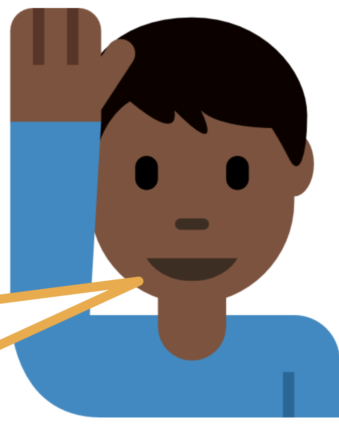
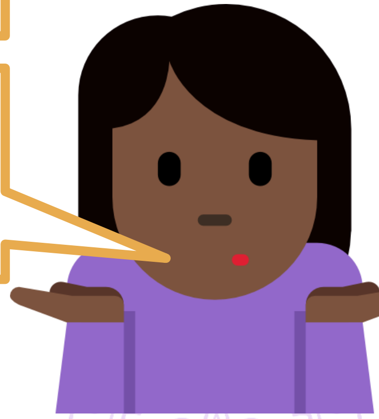
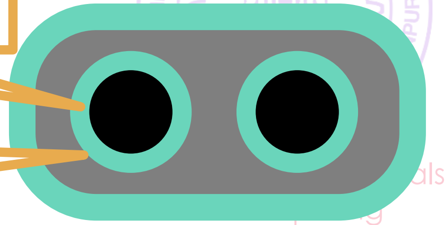
Just like int arrays, char arrays cannot be initialized this way after declaration is done

Other ways: scanf (with %s), gets

Both are very unsafe – crash!

To print: puts, printf (with %s)

getline most powerful but have to wait for it a bit ☺

The space between the two words. Space is a character too!

Hello has 5 characters, World has 5, what is the 11th character?

Partly initialized since only 11 characters this phrase

Very good!