File  Edit  View  History  Bookmarks  Tools  Help

Ascii Table - ASCII character co  X   +

← → C ⌂   🛡 🔒 www.asciitable.com   ··· ☑ ☆   🔍 Search

Most Visited  Getting Started  http://www.pgrocer.ne...  http://www.pgrocer.ne...  http://www.pgrocer.ne...  http://www.pgrocer.ne...

# www.AsciiTable.com

ASCII | Scan Codes / EBCDIC | HTML Codes | Phonetic Alphabet | Unicode v4 | Dialing Codes | Voucher Codes

## ASCII Table and Description

ASCII stands for American Standard Code for Information Interchange. Computers can only understand numbers, so an ASCII code is the numerical representation of a character such as 'a' or '@' or an action of some sort. ASCII was developed a long time ago and now the non-printing characters are rarely used for their original purpose. Below is the ASCII character table and this includes descriptions of the first 32 non-printing characters. ASCII was actually designed for use with teletypes and so the descriptions are somewhat obscure. If someone says they want your CV however in ASCII format, all this means is they want 'plain' text with no formatting such as tabs, bold or underscoring - the raw format that any computer can understand. This is usually so they can easily import the file into their own applications without issues. Notepad.exe creates ASCII text, or in MS Word you can save a file as 'text only'

| Dec | Hx | Oct | Char | | Dec | Hx | Oct | Html | Chr | Dec | Hx | Oct | Html | Chr | Dec | Hx | Oct | Html | Chr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 000 | NUL | (null) | 32 | 20 | 040 | &#32; | Space | 64 | 40 | 100 | &#64; | @ | 96 | 60 | 140 | &#96; | ` |
| 1 | 1 | 001 | SOH | (start of heading) | 33 | 21 | 041 | &#33; | ! | 65 | 41 | 101 | &#65; | A | 97 | 61 | 141 | &#97; | a |
| 2 | 2 | 002 | STX | (start of text) | 34 | 22 | 042 | &#34; | " | 66 | 42 | 102 | &#66; | B | 98 | 62 | 142 | &#98; | b |
| 3 | 3 | 003 | ETX | (end of text) | 35 | 23 | 043 | &#35; | # | 67 | 43 | 103 | &#67; | C | 99 | 63 | 143 | &#99; | c |
| 4 | 4 | 004 | EOT | (end of transmission) | 36 | 24 | 044 | &#36; | $ | 68 | 44 | 104 | &#68; | D | 100 | 64 | 144 | &#100; | d |
| 5 | 5 | 005 | ENQ | (enquiry) | 37 | 25 | 045 | &#37; | % | 69 | 45 | 105 | &#69; | E | 101 | 65 | 145 | &#101; | e |
| 6 | 6 | 006 | ACK | (acknowledge) | 38 | 26 | 046 | &#38; | & | 70 | 46 | 106 | &#70; | F | 102 | 66 | 146 | &#102; | f |
| 7 | 7 | 007 | BEL | (bell) | 39 | 27 | 047 | &#39; | ' | 71 | 47 | 107 | &#71; | G | 103 | 67 | 147 | &#103; | g |
| 8 | 8 | 010 | BS | (backspace) | 40 | 28 | 050 | &#40; | ( | 72 | 48 | 110 | &#72; | H | 104 | 68 | 150 | &#104; | h |
| 9 | 9 | 011 | TAB | (horizontal tab) | 41 | 29 | 051 | &#41; | ) | 73 | 49 | 111 | &#73; | I | 105 | 69 | 151 | &#105; | i |
| 10 | A | 012 | LF | (NL line feed, new line) | 42 | 2A | 052 | &#42; | * | 74 | 4A | 112 | &#74; | J | 106 | 6A | 152 | &#106; | j |
| 11 | B | 013 | VT | (vertical tab) | 43 | 2B | 053 | &#43; | + | 75 | 4B | 113 | &#75; | K | 107 | 6B | 153 | &#107; | k |
| 12 | C | 014 | FF | (NP form feed, new page) | 44 | 2C | 054 | &#44; | , | 76 | 4C | 114 | &#76; | L | 108 | 6C | 154 | &#108; | l |
| 13 | D | 015 | CR | (carriage return) | 45 | 2D | 055 | &#45; | - | 77 | 4D | 115 | &#77; | M | 109 | 6D | 155 | &#109; | m |
| 14 | E | 016 | SO | (shift out) | 46 | 2E | 056 | &#46; | . | 78 | 4E | 116 | &#78; | N | 110 | 6E | 156 | &#110; | n |
| 15 | F | 017 | SI | (shift in) | 47 | 2F | 057 | &#47; | / | 79 | 4F | 117 | &#79; | O | 111 | 6F | 157 | &#111; | o |
| 16 | 10 | 020 | DLE | (data link escape) | 48 | 30 | 060 | &#48; | 0 | 80 | 50 | 120 | &#80; | P | 112 | 70 | 160 | &#112; | p |
| 17 | 11 | 021 | DC1 | (device control 1) | 49 | 31 | 061 | &#49; | 1 | 81 | 51 | 121 | &#81; | Q | 113 | 71 | 161 | &#113; | q |
| 18 | 12 | 022 | DC2 | (device control 2) | 50 | 32 | 062 | &#50; | 2 | 82 | 52 | 122 | &#82; | R | 114 | 72 | 162 | &#114; | r |
| 19 | 13 | 023 | DC3 | (device control 3) | 51 | 33 | 063 | &#51; | 3 | 83 | 53 | 123 | &#83; | S | 115 | 73 | 163 | &#115; | s |
| 20 | 14 | 024 | DC4 | (device control 4) | 52 | 34 | 064 | &#52; | 4 | 84 | 54 | 124 | &#84; | T | 116 | 74 | 164 | &#116; | t |
| 21 | 15 | 025 | NAK | (negative acknowledge) | 53 | 35 | 065 | &#53; | 5 | 85 | 55 | 125 | &#85; | U | 117 | 75 | 165 | &#117; | u |
| 22 | 16 | 026 | SYN | (synchronous idle) | 54 | 36 | 066 | &#54; | 6 | 86 | 56 | 126 | &#86; | V | 118 | 76 | 166 | &#118; | v |
| 23 | 17 | 027 | ETB | (end of trans. block) | 55 | 37 | 067 | &#55; | 7 | 87 | 57 | 127 | &#87; | W | 119 | 77 | 167 | &#119; | w |
| 24 | 18 | 030 | CAN | (cancel) | 56 | 38 | 070 | &#56; | 8 | 88 | 58 | 130 | &#88; | X | 120 | 78 | 170 | &#120; | x |
| 25 | 19 | 031 | EM | (end of medium) | 57 | 39 | 071 | &#57; | 9 | 89 | 59 | 131 | &#89; | Y | 121 | 79 | 171 | &#121; | y |

🔍 Type here to search   ○   Ⱶi   ···   1:35 PM 3/20/2020

So A has a dec value of 65 and a hex value of 41.

Lets start with the hex value of 41.  If we take the hex 41 and convert it to
binary we get 0100 0001
(remember the 8421 where for 4 we turn on the 4 and for 1 we turn on the 1)
so we now know that the binary for A is 0100 0001

Now lets look at B which has a dec value of 66 and a hex value of 42.  Hex 42 is
0100 0010

C is hex 43 which is 0100 0011
I is hex 49 which is 0100 1001
J is hex 4A which is 0100 1010
K is hex 4B which is 0100 1011
O is hex 4F which is 0100 1111
P is hex 50 which is 0101 0000
Z is hex 5A which is 0101 1010

There is a pattern here.  The 010 at the beginning designates a capital letter.

**So now if I have an A I can figure out the 8 binary digits with what I know.  The first three binary digits are 010 because it is an upper case letter and the last five binary digits are 00001 because it is the first letter of the alphabet.**
**01000001**
**So now divide into groups of four and you can get the hex.  0100  0001 is 41 because in the first group of 8421 the four is turned on giving us 0100 and in the second group of 8421 the 1 is turned on giving us 0001.**

**The O is 010 and since O is the 15th letter, I have to look at 16 8 4 2 1 for the places and figure out that I need to turn on 8,4,2,1 to make 15 so I have 01111.  Putting these together I have 01001111.**

**The Z is 010 and since Z is the 26th letter, I have to look at 16 8 4 2 1 for the places and figure out that I need to turn on 16, 8, 2 to make 26 so I have 11010.  Putting these together I have 01011010.**

and this includes descriptions of the first 32 non-printing characters. ASCII was actually designed for use with teletypes and so the descriptions are somewhat obscure. If someone says they want your CV however in ASCII format, all this means is they want 'plain' text with no formatting such as tabs, bold or underscoring - the raw format that any computer can understand. This is usually so they can easily import the file into their own applications without issues. Notepad.exe creates ASCII text, or in MS Word you can save a file as 'text only'

| Dec | Hx | Oct | Char | | Dec | Hx | Oct | Html | Chr | | Dec | Hx | Oct | Html | Chr | | Dec | Hx | Oct | Html | Chr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 000 | NUL (null) | | 32 | 20 | 040 | &#32; | Space | | 64 | 40 | 100 | &#64; | @ | | 96 | 60 | 140 | &#96; | ` |
| 1 | 1 | 001 | SOH (start of heading) | | 33 | 21 | 041 | &#33; | ! | | 65 | 41 | 101 | &#65; | A | | 97 | 61 | 141 | &#97; | a |
| 2 | 2 | 002 | STX (start of text) | | 34 | 22 | 042 | &#34; | " | | 66 | 42 | 102 | &#66; | B | | 98 | 62 | 142 | &#98; | b |
| 3 | 3 | 003 | ETX (end of text) | | 35 | 23 | 043 | &#35; | # | | 67 | 43 | 103 | &#67; | C | | 99 | 63 | 143 | &#99; | c |
| 4 | 4 | 004 | EOT (end of transmission) | | 36 | 24 | 044 | &#36; | $ | | 68 | 44 | 104 | &#68; | D | | 100 | 64 | 144 | &#100; | d |
| 5 | 5 | 005 | ENQ (enquiry) | | 37 | 25 | 045 | &#37; | % | | 69 | 45 | 105 | &#69; | E | | 101 | 65 | 145 | &#101; | e |
| 6 | 6 | 006 | ACK (acknowledge) | | 38 | 26 | 046 | &#38; | & | | 70 | 46 | 106 | &#70; | F | | 102 | 66 | 146 | &#102; | f |
| 7 | 7 | 007 | BEL (bell) | | 39 | 27 | 047 | &#39; | ' | | 71 | 47 | 107 | &#71; | G | | 103 | 67 | 147 | &#103; | g |
| 8 | 8 | 010 | BS (backspace) | | 40 | 28 | 050 | &#40; | ( | | 72 | 48 | 110 | &#72; | H | | 104 | 68 | 150 | &#104; | h |
| 9 | 9 | 011 | TAB (horizontal tab) | | 41 | 29 | 051 | &#41; | ) | | 73 | 49 | 111 | &#73; | I | | 105 | 69 | 151 | &#105; | i |
| 10 | A | 012 | LF (NL line feed, new line) | | 42 | 2A | 052 | &#42; | * | | 74 | 4A | 112 | &#74; | J | | 106 | 6A | 152 | &#106; | j |
| 11 | B | 013 | VT (vertical tab) | | 43 | 2B | 053 | &#43; | + | | 75 | 4B | 113 | &#75; | K | | 107 | 6B | 153 | &#107; | k |
| 12 | C | 014 | FF (NP form feed, new page) | | 44 | 2C | 054 | &#44; | , | | 76 | 4C | 114 | &#76; | L | | 108 | 6C | 154 | &#108; | l |
| 13 | D | 015 | CR (carriage return) | | 45 | 2D | 055 | &#45; | - | | 77 | 4D | 115 | &#77; | M | | 109 | 6D | 155 | &#109; | m |
| 14 | E | 016 | SO (shift out) | | 46 | 2E | 056 | &#46; | . | | 78 | 4E | 116 | &#78; | N | | 110 | 6E | 156 | &#110; | n |
| 15 | F | 017 | SI (shift in) | | 47 | 2F | 057 | &#47; | / | | 79 | 4F | 117 | &#79; | O | | 111 | 6F | 157 | &#111; | o |
| 16 | 10 | 020 | DLE (data link escape) | | 48 | 30 | 060 | &#48; | 0 | | 80 | 50 | 120 | &#80; | P | | 112 | 70 | 160 | &#112; | p |
| 17 | 11 | 021 | DC1 (device control 1) | | 49 | 31 | 061 | &#49; | 1 | | 81 | 51 | 121 | &#81; | Q | | 113 | 71 | 161 | &#113; | q |
| 18 | 12 | 022 | DC2 (device control 2) | | 50 | 32 | 062 | &#50; | 2 | | 82 | 52 | 122 | &#82; | R | | 114 | 72 | 162 | &#114; | r |
| 19 | 13 | 023 | DC3 (device control 3) | | 51 | 33 | 063 | &#51; | 3 | | 83 | 53 | 123 | &#83; | S | | 115 | 73 | 163 | &#115; | s |
| 20 | 14 | 024 | DC4 (device control 4) | | 52 | 34 | 064 | &#52; | 4 | | 84 | 54 | 124 | &#84; | T | | 116 | 74 | 164 | &#116; | t |
| 21 | 15 | 025 | NAK (negative acknowledge) | | 53 | 35 | 065 | &#53; | 5 | | 85 | 55 | 125 | &#85; | U | | 117 | 75 | 165 | &#117; | u |
| 22 | 16 | 026 | SYN (synchronous idle) | | 54 | 36 | 066 | &#54; | 6 | | 86 | 56 | 126 | &#86; | V | | 118 | 76 | 166 | &#118; | v |
| 23 | 17 | 027 | ETB (end of trans. block) | | 55 | 37 | 067 | &#55; | 7 | | 87 | 57 | 127 | &#87; | W | | 119 | 77 | 167 | &#119; | w |
| 24 | 18 | 030 | CAN (cancel) | | 56 | 38 | 070 | &#56; | 8 | | 88 | 58 | 130 | &#88; | X | | 120 | 78 | 170 | &#120; | x |
| 25 | 19 | 031 | EM (end of medium) | | 57 | 39 | 071 | &#57; | 9 | | 89 | 59 | 131 | &#89; | Y | | 121 | 79 | 171 | &#121; | y |
| 26 | 1A | 032 | SUB (substitute) | | 58 | 3A | 072 | &#58; | : | | 90 | 5A | 132 | &#90; | Z | | 122 | 7A | 172 | &#122; | z |
| 27 | 1B | 033 | ESC (escape) | | 59 | 3B | 073 | &#59; | ; | | 91 | 5B | 133 | &#91; | [ | | 123 | 7B | 173 | &#123; | { |
| 28 | 1C | 034 | FS (file separator) | | 60 | 3C | 074 | &#60; | < | | 92 | 5C | 134 | &#92; | \ | | 124 | 7C | 174 | &#124; | | |
| 29 | 1D | 035 | GS (group separator) | | 61 | 3D | 075 | &#61; | = | | 93 | 5D | 135 | &#93; | ] | | 125 | 7D | 175 | &#125; | } |
| 30 | 1E | 036 | RS (record separator) | | 62 | 3E | 076 | &#62; | > | | 94 | 5E | 136 | &#94; | ^ | | 126 | 7E | 176 | &#126; | ~ |
| 31 | 1F | 037 | US (unit separator) | | 63 | 3F | 077 | &#63; | ? | | 95 | 5F | 137 | &#95; | _ | | 127 | 7F | 177 | &#127; | DEL |

Source: www.LookupTables.com

| Dec | Hx | Oct | Html | Chr |
|-----|-----|-----|------|-----|
| 96 | 60 | 140 | &#96; | ` |
| 97 | 61 | 141 | &#97; | a |
| 98 | 62 | 142 | &#98; | b |
| 99 | 63 | 143 | &#99; | c |
| 100 | 64 | 144 | &#100; | d |
| 101 | 65 | 145 | &#101; | e |
| 102 | 66 | 146 | &#102; | f |
| 103 | 67 | 147 | &#103; | g |
| 104 | 68 | 150 | &#104; | h |
| 105 | 69 | 151 | &#105; | i |
| 106 | 6A | 152 | &#106; | j |
| 107 | 6B | 153 | &#107; | k |
| 108 | 6C | 154 | &#108; | l |
| 109 | 6D | 155 | &#109; | m |
| 110 | 6E | 156 | &#110; | n |
| 111 | 6F | 157 | &#111; | o |
| 112 | 70 | 160 | &#112; | p |
| 113 | 71 | 161 | &#113; | q |
| 114 | 72 | 162 | &#114; | r |
| 115 | 73 | 163 | &#115; | s |
| 116 | 74 | 164 | &#116; | t |
| 117 | 75 | 165 | &#117; | u |
| 118 | 76 | 166 | &#118; | v |
| 119 | 77 | 167 | &#119; | w |
| 120 | 78 | 170 | &#120; | x |
| 121 | 79 | 171 | &#121; | y |
| 122 | 7A | 172 | &#122; | z |

Now look at lower case a which has a decimal of 97 and a hex of 61

a dec 97 and hex 61 is translated to binary 0110 0001 (note that with the 8421 the 4 and 2 are turned on)

b is hex 62 which is binary 0110 0010
i is hex 69 which is binary 0110 1001
j is hex 6A which is binary 0110 1010
o is hex 6F which is binary 0110 1111
p is hex 70 which is binary 0111 0000
z is hex 7A which is binary 0111 1010

Again we see several patterns.  The first three characters are 011 which appears in all lower case letters.

So the first three characters of the binary tell the group such as upper or lower case or numbers and the right most 5 characters are telling which one.  Notice that the last 5 are the same for A and a, for B and b, for J and j, for P and p and for Z and z.

So if you get a dump of memory in hex you can turn it into binary and the use the first three characters to tell you want group it is in and the last five characters to tell which one.
O or o is 01111 when we look at the last 5 characters so the 8,4,2 and 1 are on which adds up to 15 and o is the fifteenth letter of the alphabet
Z or z is 11010 so that means the 16 is on, the 8 is on and the 2 is on which adds up to 26 and Z is the twenty sixth letter.

| Dec | Hx | Oct | Html | Chr |
|-----|----|----|------|-----|
| 64 | 40 | 100 | &#64; | @ |
| 65 | 41 | 101 | &#65; | A |
| 66 | 42 | 102 | &#66; | B |
| 67 | 43 | 103 | &#67; | C |
| 68 | 44 | 104 | &#68; | D |
| 69 | 45 | 105 | &#69; | E |
| 70 | 46 | 106 | &#70; | F |
| 71 | 47 | 107 | &#71; | G |
| 72 | 48 | 110 | &#72; | H |
| 73 | 49 | 111 | &#73; | I |
| 74 | 4A | 112 | &#74; | J |
| 75 | 4B | 113 | &#75; | K |
| 76 | 4C | 114 | &#76; | L |
| 77 | 4D | 115 | &#77; | M |
| 78 | 4E | 116 | &#78; | N |
| 79 | 4F | 117 | &#79; | O |
| 80 | 50 | 120 | &#80; | P |
| 81 | 51 | 121 | &#81; | Q |
| 82 | 52 | 122 | &#82; | R |
| 83 | 53 | 123 | &#83; | S |
| 84 | 54 | 124 | &#84; | T |
| 85 | 55 | 125 | &#85; | U |
| 86 | 56 | 126 | &#86; | V |
| 87 | 57 | 127 | &#87; | W |
| 88 | 58 | 130 | &#88; | X |
| 89 | 59 | 131 | &#89; | Y |
| 90 | 5A | 132 | &#90; | Z |

So now look at Decimal.  A is a decimal 65 so now we need to translate that to decimal. We do that by writing the powers of 2 under each of the 8 binary bits.  See the examples below and you will see that if you translate decimal or if you transfer hex you end up with the same 8 bit binary code for the letters.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|-----|----|----|----|---|---|---|---|

This is A

| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

Z is 90 which is

| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |