

# ITFreeTraining



IPv4 addresses  
as grains of sand



IPv6 addresses  
as grains of sand

## IPv6 Addresses

For the free video please see  
<http://itfreetraining.com/ipv6/addresses>

In this video from ITFreeTraining, I will look at IPv6 addresses. IPv6 addresses are very different from IPv4 addresses and, while they look complicated at first, by the end of this video you will have an excellent understanding of how they work and they won't seem as scary as they first appear.

# Number of Addresses

- IPv6 has  $2^{128}$  addresses while IPv4 has  $2^{32}$  addresses



IPv4 addresses  
as grains of sand



IPv6 addresses  
as grains of sand

0:19 – The biggest problem with IPv4 is that there are a limited number of IP Addresses. IPv6 has 2 to the power of 128 addresses while IPv4 has 2 to the power of 32 addresses. There are a lot of different ways that people have come up with to describe the difference between the number of IPv4 and IPv6 addresses. In this video, I will use the following example and there are many more to choose from. If IP Addresses were grains of sand, IPv4 addresses would fill a truck. If IPv6 addresses were grains of sand, they would fill the sun. Quite a big difference. IPv4 addresses are running out, however IPv6 addresses should never run out.

# Format of Addresses

- Address is 128bits

2	0	0	1	:	0	d	b	8	:	0	0	b	c	:	0	0	0	0	:	0	0	0	0	:	0	a	b	c	:	1	2	3	4		
																4 bits are referred to as a nibble																			
																16 bits referred to as word, quartet																			

Previous Video  
**ITFreeTraining**



**Converting Decimals**

With the larger address space, IPv6 also comes with a new format to display these addresses. This is completely different from the way that IPv4 addresses are formatted. The IPv6 address is 128 bits. It would not be that easy to read in binary format, so IPv6 addresses are shown in hexadecimal format grouped in fours and separated by colons. An example of such an address is shown. Notice that all the hexadecimal letters are in lower case. If you are not familiar with hexadecimal, please see our previous video that explains how to convert and use hexadecimal. At first, the address looked complicated, but once you start breaking down the addresses they get a lot less complicated. Also, once you start working with them, you will start to recognize particular addresses and what they do. To start with, let's break down the address into the smallest part, which is 4 bits and referred to as a nibble. With some tasks in IPv6, like subnetting, the process is easier if the 4 bit boundary is kept, however this is not always possible. The next break down is into groups of four which is 16 bits. There are a number of different naming conventions and no one convention has been universally accepted for IPv6. In programming, 16 bits is referred to as a word. In this course, 16 bits or a block of four characters in an address will be referred to as a quartet. Just be mindful that other naming conventions exist and it may be a while before one is agreed upon, if at all. The IPv6 address is a little long and complicated, so let's look at how it can be made easier to work with.

# Zero Compression

Leading zeros can be removed

2001:0db8:00bc:0000:0000:0000:0abc:1234

2001: db8: bc:0000:0000:0000: abc:1234

2001:db8:bc:0000:0000:0000:abc:1234

3:00 – To make the IPv6 addresses a little easier to read, first, the leading zeroes in each group of four can be removed. To demonstrate this, I will use the previous example. By looking at the address, notice that there are a number of leading zeroes. These leading zeros can be removed to give the following. You would not display the address like this, so the next step would be to remove the white space. This will give a smaller address that is easier to work with. There are, however, a lot of zeroes left in the address and it is possible to make it even smaller. Let's have a look how.

# Consecutive Zeros

Consecutive zeros can be removed and replaced with ::

2001 : db8 : bc : 0000 : 0000 : 0000 : abc : 1234

2001 : db8 : bc : : : : abc : 1234

2001 : db8 : bc :: abc : 1234

It is possible to compress an IPv6 address by removing consecutive zeroes and replacing them with a double colon. This is the example IPv6 address from the previous slide and we have three sets of consecutive zeroes. These can actually be removed and replaced with one double colon. As you can see with the example, the zeroes have been removed and there is white space between them. Once the white space is removed, you can see the actual physical address at the bottom. You can see that this address is much easier to read and recall now that it has had all the excess zeroes removed. In some cases, however, you will have two sets of consecutive zeroes so let's take a look at that next.

## Consecutive Zeros Rule

Consecutive zeros can only be removed once

2001:db8:bc:0000:0000:abc:0000:0000



2001:db8:bc::abc:0000:0000 

2001:db8:bc:0000:0000:abc:0000:0000



2001:db8:bc:0000:0000:abc:: 

2001:db8:bc:0000:0000:abc:0000:0000



2001:db8:bc::abc:: 

4:31 – With IPv6, consecutive zeroes in an address can be replaced with a double colon in order to further compress the address and continue to help make it easier to read. Here we'll take a look at an example of replacing the consecutive zeroes with colons. The consecutive zeroes can be anywhere in the address, but this replacement of the zeroes with a colon can only be done once. In the first case, the first group of zeroes can be reduced to a double colon. Consider the example again; this time the second group of zeroes will be reduced. This will give the following result, which is valid. The point to remember is that both are valid as they use a double colon only once. The last example, removing two sets of zeroes, is incorrect and not acceptable in IPv6. This is because it is not possible to expand it and get only one result. Looking at this third incorrect example, we could get one set of zeroes in the middle and the three at the end, or three sets of zeroes and one at the end leaving us with multiple possible solutions. This is why the replacement of consecutive zeroes can only happen once.

# Loopback Address Compressing

- Loopback address is
  - 0000:0000:0000:0000:0000:0000:0000:0001
- Compresses to
  - ::1

5:44 – Later in the course I cover the loopback address in more detail, but I will mention it here because it is a good example of zero compression. The loopback address is seven groups of four zeroes or seven quartets of zeroes followed by three zeroes and a one. Not something you want to be typing in if you can avoid it. Using zero compression, the address can be reduced to two colons and a one. Much easier to type in. You can see that using zero compression makes the administrator's job a lot easier.

# Ports

- [IPv6 Address]:port
  - [2001:db8:bc::abc:1234] : 8080
  - http:// [2001:db8:bc::abc:1234] : 8080

6:20 – The next point I would like to look at is, how would a port be added to an IPv6 address? In order to do this, the IPv6 address is encased in square brackets, a colon added, then followed by the port. If for example, the address was being used in a troubleshooting command like ping, the following would be used “[2001:db8:bc::abc:1234]:8080”. If for example a URL was being used, the same process applies. The square brackets are required because if zero compression was used, it would not be possible to tell the difference between the address and the port number. For example, the 8080 shown here without square brackets could be interpreted as part of the IPv6 address rather than a port number.



# Network Prefix

- Essentially the same as CIDR in IPv4
  - `2001:0db8:00bc:0000:0000:0000:0abc:1234/64`
  - `2002::/16`
- Mostly used in routing
- Computers will almost always use 64

7:05 – The next topic I will look at is the network prefix. The network prefix is essentially the same as Classless Inter-Domain Routing or CIDR, that is used in IPv4. You can see the following example of an IPv6 address with a slash 64 on the end. This is the same notation that IPv4 uses. However, as we have learned, unlike IPv4 the addresses can be shortened. So, for example, the following would also be valid “2001:0db8:00bc:0000:0000:0000:0abc:1234/64”. Unlike IPv4, in IPv6 the network prefixes are mostly used in routing. In IPv4, the administrator would traditionally configure a subnet mask on a computer on a network, like a workstation or a server. However, in IPv6, in almost all cases, a computer on the network will always have a network prefix of 64. In later videos, I will go into more detail why this is the case. However, for the moment, if you are in doubt use the network prefix of 64.

# Zone IDs

- Defines which interface should be used
  - E.g. Ping 2001:db8:bc::abc:0:1%2



8:06 – The last topic that I will go into in this video is Zone IDs. When a Zone ID is used with an IPv6 address, it defines which interface should be used. In Windows, generally the Zone ID will be the same as the interface ID. In some rare cases, you may want to do this. For example, if I were to ping the following IPv6 address “Ping 2001:db8:bc::abc:0:1%2”, I could add %2 to the end of the IP Address to ensure that the ping command sends data out of the second interface. In some cases, in IPv6 you will need to add the Zone ID to an IPv6 address to ensure the correct interface is used. At least now, if you see the Zone ID used with an IPv6 address, you will understand what it means. In later videos, I will go into this topic in more detail. That covers it for IPv6 addresses. You should now have a basic understanding of what an IPv6 address looks like. They still, most likely, look long and complicated. In later videos, I will break them down even more and you will start to recognize different IPv6 addresses when you see them which will make working with them a lot easier. I hope to see you in those videos.