)init	Initializes the receiver, a newly allocated NSString, to contain no characters. This is the only initialization method that a subclass of NSString should invoke.
)initWithCString:(const char *)byteString	Initializes the receiver, a newly allocated NSString, by converting the one-byte characters in byteString into Unicode characters. byteString must be a null-terminated C string in the default C string encoding.
)initWithCString:(const char *)byteString length:(unsigned int)length	Initializes the receiver, a newly allocated NSString, by converting length one-byte characters in byteString into Unicode characters. This method doesn't stop at a null byte.
)initWithCStringNoCopy:(char *)byteString length:(unsigned int)length freeWhenDone:(BOOL)flag	Initializes the receiver, a newly allocated NSString, by converting length one-byte characters in byteString into Unicode characters. This method doesn't stop at a null byte. The receiver

	but if flag is NO it won't. Note that the NO case could with memory that could be freed. The NO flag should provided backing store is permanent.
)initWithContentsOfFile:(NSString *)path	Initializes the receiver, a newly allocated NSString, by rea the file whose name is given by path. This method atte encoding for the file. The string is assumed to be in Un the encoding is determined not to be Unicode, the defa used instead. Also see writeToFile:atomically: in [a]Stor
)initWithData:(NSData *)data encoding:(NSStringEncoding)encoding	Initializes the receiver, a newly allocated NSString, by converting the bytes in data into Unicode characters. d object containing bytes in encoding and in the default [a] that encoding.
)initWithFormat:(NSString *)format,...	Initializes the receiver, a newly allocated NSString, by co format and following string objects in the manner of pr
)initWithFormat:(NSString *)format arguments:(va_list)argList	Initializes the receiver, a newly allocated NSString, by constructing a string from format and argList in the ma
)initWithFormat:(NSString *)format locale:(NSDictionary *)dictionary,...	Initializes the receiver, a newly allocated NSString, by constructing a string from format and the formatting in dictionary in the manner of printf().
)initWithFormat:(NSString *)format locale:(NSDictionary *)dictionary arguments:(va_list)argList	Initializes the receiver, a newly allocated NSString, by constructing a string from format and format information in dictionary and argList in the manner of
)initWithString:(NSString *)string	Initializes the receiver, a newly allocated NSString, by co from string.
signed int)length	Returns the number of characters in the receiver. This num individual characters of composed character sequences
ichar)characterAtIndex:(unsigned int)index	Returns the character at the array position given by index. NSStringBoundsError exception if index lies beyond th
id)getCharacters:(unichar *)buffer	Invokes getCharacters:range: with the provided buffer and receiver as the range.
id)getCharacters:(unichar *)buffer range:(NSRange)aRange	Copies characters from aRange in the receiver into buffer, which must be large enough to contain them. This met character. This method raises an NSStringBoundsError of aRange lies beyond the end of the string.
SString *)stringByAppendingFormat:(NSString *)format,...	Returns a string made by using format as a printf() style fo following arguments as values to be substituted into th
SString *)stringByAppendingString:(NSString *)aString	Returns a string made by appending aString and the receiv

	one at index to the end. This method raises an NSStrin exception if index lies beyond the end of the string.
SString *)substringFromRange:(NSRange)aRange	Returns a string object containing the characters of the rec aRange. This method raises an NSStringBoundsError e aRange lies beyond the end of the string.
SString *)substringToIndex:(unsigned int)index	Returns a string object containing the characters of the rec including, the one at index. This method raises an NSS exception if index lies beyond the end of the string.
SRange)rangeOfCharacterFromSet:(NSCharacterSet *)aSet	Invokes rangeOfCharacterFromSet:options: with no optio
SRange)rangeOfCharacterFromSet:(NSCharacterSet *)aSet options:(unsigned int)mask	Invokes rangeOfCharacterFromSet:options:range: with ma of the receiver as the range.
SRange)rangeOfCharacterFromSet:(NSCharacterSet *)aSet options:(unsigned int)mask range:(NSRange)aRange	Returns the range of the first character found from aSet. The search is restricted to aRange with mask options. m combination (using the C bitwise OR operator |) of NS NSLiteralSearch, and NSBackwardsSearch.
SRange)rangeOfString:(NSString *)string	Invokes rangeOfString:options: with no options.
SRange)rangeOfString:(NSString *)string options:(unsigned int)mask	Invokes rangeOfString:options:range: with mask options and the entire extent of the receiver as the rang
SRange)rangeOfString:(NSString *)aString options:(unsigned int)mask range:(NSRange)aRange	Returns the range giving the location and length in the receiver of aString. The search is restricted to aRange with mask options. mask can be any combination (usin operator |) of NSCaseInsensitiveSearch, NSLiteralSear NSBackwardsSearch, and NSAnchoredSearch.
SRange)rangeOfComposedCharacterSequenceAtIndex:(unsigned int)anIndex	Returns an NSRange giving the location and length in the composed character sequence located at anIndex. This NSStringBoundsError exception if anIndex lies beyon
SComparisonResult)caseInsensitiveCompare:(NSString *)aString	Invokes compare:options: with the option NSCaseInsensit
SComparisonResult)compare:(NSString *)aString	Invokes compare:options: with no options.
SComparisonResult)compare:(NSString *)aString options:(unsigned int)mask	Invokes compare:options:range: with mask as the options extent as the range.
SComparisonResult)compare:(NSString *)aString	

	structure. If two string objects are equal (as determined method), they must have the same hash value.
OOL)isEqual:(id)anObject	Returns YES if both the receiver and anObject have the sa NSStrings that compare as NSOrderedSame, NO other
OOL)isEqualToString:(NSString *)aString	Returns YES if aString is equivalent to the receiver (if the they compare as NSOrderedSame), NO otherwise.
SString *)description	Returns the string itself.
OOL)writeToFile:(NSString *)filename atomically:(BOOL)useAuxiliaryFile	Writes a textual description of the receiver to filename. If useAuxiliaryFile is YES, the data is written to a bac assuming no errors occur, the backup file is renamed to name. The string is written in the default C string enco be converted to that encoding. If not, the string is store encoding.
SString *)commonPrefixWithString:(NSString *)aString options:(unsigned int)mask	Returns the substring of the receiver containing characters aString have in common. mask can be any combinatio OR operator |) of NSCaseInsensitiveSearch and NSLit
SString *)capitalizedString	Returns a string with the first character of each word chan corresponding uppercase value.
SString *)lowercaseString	Returns a string with each character changed to its corresp value.
SString *)uppercaseString	Returns a string with each character changed to its corresp value.
nst char *)cString	Returns a representation of the receiver as a C string in the encoding.
signed int)cStringLength	Returns the length in bytes of the C string representation o
id)getCString:(char *)buffer	Invokes getCString:maxLength:range:remainingRange: w NSMaximumStringLength as the maximum length, the extent as the range, and NULL for the remaining range enough to contain the resulting C string plus a terminat (which this method adds).
id)getCString:(char *)buffer maxLength:(unsigned int)maxLength	Invokes getCString:maxLength:range:remainingRange: with m maximum length, the receiver's entire extent as the ra remaining range. buffer must be large enough to contai plus a terminating null character (which this method ad
id)getCString:(char *)buffer	Copies the receiver's characters (in the default C string

	Whitespace at the beginning of the string is skipped. If with a valid text representation of a floating-point num value is returned, otherwise 0.0 is returned. HUGE_VA returned on overflow. 0.0 is returned on underflow. Ch number are ignored.
oat)floatValue	Returns the floating-point value of the receiver's text. Wh beginning of the string is skipped. If the receiver begin representation of a floating-point number, that number otherwise 0.0 is returned. HUGE_VAL or HUGE_VA overflow. 0.0 is returned on underflow. Characters foll ignored.
t)intValue	Returns the integer value of the receiver's text. Whitespac the string is skipped. If the receiver begins with a valid integer, that number's value is returned, otherwise 0 is INT_MIN is returned on overflow. Characters followi ignored.
OOL)canBeConvertedToEncoding:(NSStringEncoding)encoding	Returns YES if the receiver can be converted to encoding information, and NO otherwise.
SData *)dataUsingEncoding:(NSStringEncoding)encoding	Invokes dataUsingEncoding:allowLossyConversion: with allow lossy conversion.
SData *)dataUsingEncoding:(NSStringEncoding)encoding allowLossyConversion:(BOOL)flag	Returns an NSData object containing a representation of t If flag is NO and the receiver can't be converted witho information (such as accents or case) this method retur and the receiver can't be converted without losing som characters may be removed or altered in conversion.
SStringEncoding)fastestEncoding	Encoding in which this string can be expressed (with loss quickly.
SStringEncoding)smallestEncoding	Encoding in which this string can be expressed (with lossl most space efficient manner
)propertyList	Depending on the format of the receiver's contents, return or dictionary object represention of those contents.
SDictionary *)propertyListFromStringsFileFormat	Returns a dictionary object initialized with the keys and va receiver. The receiver's format must be that used for [a].

ª/ Images/Bloggs.tiffº, this method returns a string cont

SString *)pathExtension — Returns the extension of the receiver's path representation Images/Bloggs.tiffº, this method returns a string conta

SString *)stringByAbbreviatingWithTildeInPath

Returns a string in which the user's home directory path is

SString *)stringByAppendingPathComponent:(NSString *)aString

Returns a string representing the receiver's path with the a component aString.

SString *)stringByAppendingPathExtension:(NSString *)aString

Returns a string representing the receiver's path with the a extension aString.

SString *)stringByDeletingLastPathComponent

Returns the receiver's path representation minus the last c path ª/ Images/Bloggs.tiffº, this method returns a string

SString *)stringByDeletingPathExtension — Returns the receiver's path representation minus the exten component. Given the path ª/ Images/Bloggs.tiffº, this containing ª/ Images/Bloggsº.

SString *)stringByExpandingTildeInPath — Returns a string in which a tilde is expanded to its full pat

SString *)stringByResolvingSymlinksInPath — Returns a string identical to the receiver's path except that have been resolved.

SString *)stringByStandardizingPath — Returns a string containing a ªstandardizedº path, one in w expanded and redundant elements (for example ª/ /º) el