

Python Programming Fundamentals Cheat Sheet		
Package/Method	Description	Syntax and Code Example
AND	Returns 'True' if both statement1 and statement2 are 'True'. Otherwise, returns 'False'.	<p>Syntax:</p> <pre>1. 1 2. 1. statement1 and statement2</pre> <p>Copy</p> <p>Example:</p> <pre>1. 1 2. 2 3. 2 4. 2 5. 2 6. 2 7. 2 8. 2</pre> <pre>1. marks = 88 2. attendance_percentage = 97 3. if marks >= 88 and attendance_percentage >= 80: 4. print("qualify for honors") 5. else: 6. print("not qualify for honors") 7. # Output - qualify for honors</pre> <p>Copy</p>
Class Definition	Defines a blueprint for creating objects and defining their attributes and behaviors.	<p>Syntax:</p> <pre>1. 1 2. 1. class ClassName: # Class attributes and methods</pre> <p>Copy</p> <p>Example:</p> <pre>1. 1 2. 2 3. 2 4. 2</pre> <pre>1. class Person: 2. def __init__(self, name, age): 3. self.name = name 4. self.age = age</pre> <p>Copy</p>
Define Function	A 'function' is a reusable block of code that performs a specific task or set of tasks when called.	<p>Syntax:</p> <pre>1. 1 2. 1. def function_name(parameters): # function body</pre> <p>Copy</p> <p>Example:</p> <pre>1. 1 2. 1. def greet(name): print("Hello,", name)</pre> <p>Copy</p>
Equal(==)	Checks if two values are equal.	<p>Syntax:</p> <pre>1. 1 2. 1. variable1 == variable2</pre> <p>Copy</p> <p>Example 1:</p> <pre>1. 1 2. 1. 5 == 5</pre> <p>Copy</p> <p>returns True</p> <p>Example 2:</p> <pre>1. 1 2. 1. age = 25 age == 30</pre> <p>Copy</p> <p>returns False</p> <p>Syntax:</p> <pre>1. 1 2. 1. for variable in sequence: # Code to repeat</pre> <p>Copy</p>
For Loop	A 'for' loop repeatedly executes a block of code for a specified number of iterations or over a sequence of elements (list, range, string, etc.).	<p>Example 1:</p> <pre>1. 1 2. 1. for num in range(1, 10): 3. print(num)</pre> <p>Copy</p> <p>Example 2:</p> <pre>1. 1 2. 2 3. 2 4. 2</pre> <pre>1. fruits = ["apple", "banana", "orange", "grape", "chic"] 2. for fruit in fruits: 3. print(fruit)</pre> <p>Copy</p> <p>Syntax:</p> <pre>1. 1 2. 1. function_name(arguments)</pre> <p>Copy</p> <p>Example:</p> <pre>1. 1 2. 1. greet("Alice")</pre> <p>Copy</p> <p>Syntax:</p> <pre>1. 1 2. 1. variable1 <= variable2</pre> <p>Copy</p>
Function Call	A function call is the act of executing the code within the function using the provided arguments.	<p>Example 1:</p> <pre>1. 1 2. 1. 5 > 5 and 5 > 5</pre> <p>Copy</p> <p>returns True</p> <p>Example 2:</p> <pre>1. 1 2. 2 3. 2 4. 2</pre> <pre>1. quantity = 100 2. quantity > 100 3. quantity <= 100</pre> <p>Copy</p> <p>returns True</p> <p>Syntax:</p> <pre>1. 1 2. 1. variable1 > variable2</pre> <p>Copy</p>
Greater Than or Equal To(>=)	Checks if the value of variable1 is greater than or equal to variable2.	<p>Example 1: 9 >= 6</p> <p>returns True</p> <p>Example 2:</p> <pre>1. 1 2. 2 3. 2 4. 2</pre> <pre>1. num = 10 2. num >= 10 3. num <= 10, num_age</pre> <p>Copy</p> <p>returns False</p> <p>Syntax:</p> <pre>1. 1 2. 1. if condition: #code block for if statement</pre> <p>Copy</p> <p>Example:</p> <pre>1. 1 2. 2 3. 2 4. 2</pre> <pre>1. if temperature > 100: 2. print("It's a hot day!")</pre> <p>Copy</p> <p>Syntax:</p> <pre>1. 1 2. 2 3. 2 4. 2 5. 2 6. 2</pre> <pre>1. if condition1: 2. # code if condition1 is True 3. elif condition2: 4. # code if condition2 is True 5. else: 6. # code if no condition is True</pre> <p>Copy</p>
Greater Than(>)	Checks if the value of variable1 is greater than variable2.	<p>Example:</p> <pre>1. 1 2. 2 3. 2 4. 2 5. 2 6. 2 7. 2 8. 2</pre> <pre>1. score = 80 # Example score 2. if score >= 80: 3. print("You got an A!") 4. elif score >= 70: 5. print("You got a B.") 6. else: 7. print("You need to work harder.") 8. # Output - You got a B.</pre> <p>Copy</p> <p>Syntax:</p> <pre>1. 1 2. 2 3. 2</pre> <pre>1. if condition: # code, if condition is True 2. else: # code, if condition is False</pre> <p>Copy</p>
If Statement	Executes code block 'if' the condition is 'True'.	<p>Example:</p> <pre>1. 1 2. 2 3. 2 4. 2</pre> <pre>1. if age >= 18: 2. print("You're an adult.") 3. else: 4. print("You're not an adult yet.")</pre> <p>Copy</p> <p>Syntax:</p> <pre>1. 1 2. 1. variable1 <= variable2</pre> <p>Copy</p>
If Else Else	Executes the first code block if conditional is 'True', otherwise checks condition2, and so on. If no condition is 'True', the else block is executed.	<p>Example:</p> <pre>1. 1 2. 2 3. 2 4. 2 5. 2 6. 2 7. 2 8. 2</pre> <pre>1. score = 80 # Example score 2. if score >= 80: 3. print("You got an A!") 4. elif score >= 70: 5. print("You got a B.") 6. else: 7. print("You need to work harder.") 8. # Output - You got a B.</pre> <p>Copy</p> <p>Syntax:</p> <pre>1. 1 2. 2 3. 2</pre> <pre>1. if condition: # code, if condition is True 2. else: # code, if condition is False</pre> <p>Copy</p>
If Else Statement	Executes the first code block if the condition is 'True', otherwise the second block.	<p>Example:</p> <pre>1. 1 2. 2 3. 2 4. 2</pre> <pre>1. if age >= 18: 2. print("You're an adult.") 3. else: 4. print("You're not an adult yet.")</pre> <p>Copy</p> <p>Syntax:</p> <pre>1. 1 2. 1. variable1 <= variable2</pre> <p>Copy</p>
Less Than or Equal To(<=)	Checks if the value of variable1 is less than or equal to variable2.	<p>Example 1:</p> <pre>1. 1 2. 1. 5 <= 5 and 5 <= 5</pre> <p>Copy</p>

		<div><div>return True</div><div>Example 2:</div><div><pre>1. 1 2. 2 3. 3 4. else: 30 5. else: else = 40 6. else: else = else_size</pre></div><div>Copy</div><div>return True</div><div>System:</div><div><pre>1. 1 2. variable1 = variable1</pre></div><div>Copy</div><div>Example 1:</div><div><pre>1. 1 2. 1 < 4</pre></div><div>Copy</div></div>
Less Than(<)	Checks if the value of variable1 is less than variable2.	<div><div>return True</div><div>Example 2:</div><div><pre>1. 1 2. 2 3. 3 4. score = 40 5. missing_score = 40 6. score = missing_score</pre></div><div>Copy</div><div>return True</div><div>System:</div><div><pre>1. 1 2. 1 3. 2 4. 3 5. 4 6. 5 7. 6 8. 7 9. 8 10. 9 11. 10 12. 11 13. 12 14. 13 15. 14 16. 15 17. 16 18. 17 19. 18 20. 19 21. 20 22. 21 23. 22 24. 23 25. 24 26. 25 27. 26 28. 27 29. 28 30. 29 31. 30 32. 31 33. 32 34. 33 35. 34 36. 35 37. 36 38. 37 39. 38 40. 39 41. 40 42. 41 43. 42 44. 43 45. 44 46. 45 47. 46 48. 47 49. 48 50. 49 51. 50 52. 51 53. 52 54. 53 55. 54 56. 55 57. 56 58. 57 59. 58 60. 59 61. 60 62. 61 63. 62 64. 63 65. 64 66. 65 67. 66 68. 67 69. 68 70. 69 71. 70 72. 71 73. 72 74. 73 75. 74 76. 75 77. 76 78. 77 79. 78 80. 79 81. 80 82. 81 83. 82 84. 83 85. 84 86. 85 87. 86 88. 87 89. 88 90. 89 91. 90 92. 91 93. 92 94. 93 95. 94 96. 95 97. 96 98. 97 99. 98 100. 99 101. 100 102. 101 103. 102 104. 103 105. 104 106. 105 107. 106 108. 107 109. 108 110. 109 111. 110 112. 111 113. 112 114. 113 115. 114 116. 115 117. 116 118. 117 119. 118 120. 119 121. 120 122. 121 123. 122 124. 123 125. 124 126. 125 127. 126 128. 127 129. 128 130. 129 131. 130 132. 131 133. 132 134. 133 135. 134 136. 135 137. 136 138. 137 139. 138 140. 139 141. 140 142. 141 143. 142 144. 143 145. 144 146. 145 147. 146 148. 147 149. 148 150. 149 151. 150 152. 151 153. 152 154. 153 155. 154 156. 155 157. 156 158. 157 159. 158 160. 159 161. 160 162. 161 163. 162 164. 163 165. 164 166. 165 167. 166 168. 167 169. 168 170. 169 171. 170 172. 171 173. 172 174. 173 175. 174 176. 175 177. 176 178. 177 179. 178 180. 179 181. 180 182. 181 183. 182 184. 183 185. 184 186. 185 187. 186 188. 187 189. 188 190. 189 191. 190 192. 191 193. 192 194. 193 195. 194 196. 195 197. 196 198. 197 199. 198 200. 199 201. 200 202. 201 203. 202 204. 203 205. 204 206. 205 207. 206 208. 207 209. 208 210. 209 211. 210 212. 211 213. 212 214. 213 215. 214 216. 215 217. 216 218. 217 219. 218 220. 219 221. 220 222. 221 223. 222 224. 223 225. 224 226. 225 227. 226 228. 227 229. 228 230. 229 231. 230 232. 231 233. 232 234. 233 235. 234 236. 235 237. 236 238. 237 239. 238 240. 239 241. 240 242. 241 243. 242 244. 243 245. 244 246. 245 247. 246 248. 247 249. 248 250. 249 251. 250 252. 251 253. 252 254. 253 255. 254 256. 255 257. 256 258. 257 259. 258 260. 259 261. 260 262. 261 263. 262 264. 263 265. 264 266. 265 267. 266 268. 267 269. 268 270. 269 271. 270 272. 271 273. 272 274. 273 275. 274 276. 275 277. 276 278. 277 279. 278 280. 279 281. 280 282. 281 283. 282 284. 283 285. 284 286. 285 287. 286 288. 287 289. 288 290. 289 291. 290 292. 291 293. 292 294. 293 295. 294 296. 295 297. 296 298. 297 299. 298 300. 299 301. 300 302. 301 303. 302 304. 303 305. 304 306. 305 307. 306 308. 307 309. 308 310. 309 311. 310 312. 311 313. 312 314. 313 315. 314 316. 315 317. 316 318. 317 319. 318 320. 319 321. 320 322. 321 323. 322 324. 323 325. 324 326. 325 327. 326 328. 327 329. 328 330. 329 331. 330 332. 331 333. 332 334. 333 335. 334 336. 335 337. 336 338. 337 339. 338 340. 339 341. 340 342. 341 343. 342 344. 343 345. 344 346. 345 347. 346 348. 347 349. 348 350. 349 351. 350 352. 351 353. 352 354. 353 355. 354 356. 355 357. 356 358. 357 359. 358 360. 359 361. 360 362. 361 363. 362 364. 363 365. 364 366. 365 367. 366 368. 367 369. 368 370. 369 371. 370 372. 371 373. 372 374. 373 375. 374 376. 375 377. 376 378. 377 379. 378 380. 379 381. 380 382. 381 383. 382 384. 383 385. 384 386. 385 387. 386 388. 387 389. 388 390. 389 391. 390 392. 391 393. 392 394. 393 395. 394 396. 395 397. 396 398. 397 399. 398 400. 399 401. 400 402. 401 403. 402 404. 403 405. 404 406. 405 407. 406 408. 407 409. 408 410. 409 411. 410 412. 411 413. 412 414. 413 415. 414 416. 415 417. 416 418. 417 419. 418 420. 419 421. 420 422. 421 423. 422 424. 423 425. 424 426. 425 427. 426 428. 427 429. 428 430. 429 431. 430 432. 431 433. 432 434. 433 435. 434 436. 435 437. 436 438. 437 439. 438 440. 439 441. 440 442. 441 443. 442 444. 443 445. 444 446. 445 447. 446 448. 447 449. 448 450. 449 451. 450 452. 451 453. 452 454. 453 455. 454 456. 455 457. 456 458. 457 459. 458 460. 459 461. 460 462. 461 463. 462 464. 463 465. 464 466. 465 467. 466 468. 467 469. 468 470. 469 471. 470 472. 471 473. 472 474. 473 475. 474 476. 475 477. 476 478. 477 479. 478 480. 479 481. 480 482. 481 483. 482 484. 483 485. 484 486. 485 487. 486 488. 487 489. 488 490. 489 491. 490 492. 491 493. 492 494. 493 495. 494 496. 495 497. 496 498. 497 499. 498 500. 499 501. 500 502. 501 503. 502 504. 503 505. 504 506. 505 507. 506 508. 507 509. 508 510. 509 511. 510 512. 511 513. 512 514. 513 515. 514 516. 515 517. 516 518. 517 519. 518 520. 519 521. 520 522. 521 523. 522 524. 523 525. 524 526. 525 527. 526 528. 527 529. 528 530. 529 531. 530 532. 531 533. 532 534. 533 535. 534 536. 535 537. 536 538. 537 539. 538 540. 539 541. 540 542. 541 543. 542 544. 543 545. 544 546. 545 547. 546 548. 547 549. 548 550. 549 551. 550 552. 551 553. 552 554. 553 555. 554 556. 555 557. 556 558. 557 559. 558 560. 559 561. 560 562. 561 563. 562 564. 563 565. 564 566. 565 567. 566 568. 567 569. 568 570. 569 571. 570 572. 571 573. 572 574. 573 575. 574 576. 575 577. 576 578. 577 579. 578 580. 579 581. 580 582. 581 583. 582 584. 583 585. 584 586. 585 587. 586 588. 587 589. 588 590. 589 591. 590 592. 591 593. 592 594. 593 595. 594 596. 595 597. 596 598. 597 599. 598 600. 599 601. 600 602. 601 603. 602 604. 603 605. 604 606. 605 607. 606 608. 607 609. 608 610. 609 611. 610 612. 611 613. 612 614. 613 615. 614 616. 615 617. 616 618. 617 619. 618 620. 619 621. 620 622. 621 623. 622 624. 623 625. 624 626. 625 627. 626 628. 627 629. 628 630. 629 631. 630 632. 631 633. 632 634. 633 635. 634 636. 635 637. 636 638. 637 639. 638 640. 639 641. 640 642. 641 643. 642 644. 643 645. 644 646. 645 647. 646 648. 647 649. 648 650. 649 651. 650 652. 651 653. 652 654. 653 655. 654 656. 655 657. 656 658. 657 659. 658 660. 659 661. 660 662. 661 663. 662 664. 663 665. 664 666. 665 667. 666 668. 667 669. 668 670. 669 671. 670 672. 671 673. 672 674. 673 675. 674 676. 675 677. 676 678. 677 679. 678 680. 679 681. 680 682. 681 683. 682 684. 683 685. 684 686. 685 687. 686 688. 687 689. 688 690. 689 691. 690 692. 691 693. 692 694. 693 695. 694 696. 695 697. 696 698. 697 699. 698 700. 699 701. 700 702. 701 703. 702 704. 703 705. 704 706. 705 707. 706 708. 707 709. 708 710. 709 711. 710 712. 711 713. 712 714. 713 715. 714 716. 715 717. 716 718. 717 719. 718 720. 719 721. 720 722. 721 723. 722 724. 723 725. 724 726. 725 727. 726 728. 727 729. 728 730. 729 731. 730 732. 731 733. 732 734. 733 735. 734 736. 735 737. 736 738. 737 739. 738 740. 739 741. 740 742. 741 743. 742 744. 743 745. 744 746. 745 747. 746 748. 747 749. 748 750. 749 751. 750 752. 751 753. 752 754. 753 755. 754 756. 755 757. 756 758. 757 759. 758 760. 759 761. 760 762. 761 763. 762 764. 763 765. 764 766. 765 767. 766 768. 767 769. 768 770. 769 771. 770 772. 771 773. 772 774. 773 775. 774 776. 775 777. 776 778. 777 779. 778 780. 779 781. 780 782. 781 783. 782 784. 783 785. 784 786. 785 787. 786 788. 787 789. 788 790. 789 791. 790 792. 791 793. 792 794. 793 795. 794 796. 795 797. 796 798. 797 799. 798 800. 799 801. 800 802. 801 803. 802 804. 803 805. 804 806. 805 807. 806 808. 807 809. 808 810. 809 811. 810 812. 811 813. 812 814. 813 815. 814 816. 815 817. 816 818. 817 819. 818 820. 819 821. 820 822. 821 823. 822 824. 823 825. 824 826. 825 827. 826 828. 827 829. 828 830. 829 831. 830 832. 831 833. 832 834. 833 835. 834 836. 835 837. 836 838. 837 839. 838 840. 839 841. 840 842. 841 843. 842 844. 843 845. 844 846. 845 847. 846 848. 847 849. 848 850. 849 851. 850 852. 851 853. 852 854. 853 855. 854 856. 855 857. 856 858. 857 859. 858 860. 859 861. 860 862. 861 863. 862 864. 863 865. 864 866. 865 867. 866 868. 867 869. 868 870. 869 871. 870 872. 871 873. 872 874. 873 875. 874 876. 875 877. 876 878. 877 879. 878 880. 879 881. 880 882. 881 883. 882 884. 883 885. 884 886. 885 887. 886 888. 887 889. 888 890. 889 891. 890 892. 891 893. 892 894. 893 895. 894 896. 895 897. 896 898. 897 899. 898 900. 899 901. 900 902. 901 903. 902 904. 903 905. 904 906. 905 907. 906 908. 907 909. 908 910. 909 911. 910 912. 911 913. 912 914. 913 915. 914 916. 915 917. 916 918. 917 919. 918 920. 919 921. 920 922. 921 923. 922 924. 923 925. 924 926. 925 927. 926 928. 927 929. 928 930. 929 931. 930 932. 931 933. 932 934. 933 935. 934 936. 935 937. 936 938. 937 939. 938 940. 939 941. 940 942. 941 943. 942 944. 943 945. 944 946. 945 947. 946 948. 947 949. 948 950. 949 951. 950 952. 951 953. 952 954. 953 955. 954 956. 955 957. 956 958. 957 959. 958 960. 959 961. 960 962. 961 963. 962 964. 963 965. 964 966. 965 967. 966 968. 967 969. 968 970. 969 971. 970 972. 971 973. 972 974. 973 975. 974 976. 975 977. 976 978. 977 979. 978 980. 979 981. 980 982. 981 983. 982 984. 983 985. 984 986. 985 987. 986 988. 987 989. 988 990. 989 991. 990 992. 991 993. 992 994. 993 995. 994 996. 995 997. 996 998. 997 999. 998 1000. 999 1001. 1000 1002. 1001 1003. 1002 1004. 1003 1005. 1004 1006. 1005 1007. 1006 1008. 1007 1009. 1008 1010. 1009 1011. 1010 1012. 1011 1013. 1012 1014. 1013 1015. 1014 1016. 1015 1017. 1016 1018. 1017 1019. 1018 1020. 1019 1021. 1020 1022. 1021 1023. 1022 1024. 1023 1025. 1024 1026. 1025 1027. 1026 1028. 1027 1029. 1028 1030. 1029 1031. 1030 1032. 1031 1033. 1032 1034. 1033 1035. 1034 1036. 1035 1037. 1036 1038. 1037 1039. 1038 1040. 1039 1041. 1040 1042. 1041 1043. 1042 1044. 1043 1045. 1044 1046. 1045 1047. 1046 1048. 1047 1049. 1048 1050. 1049 1051. 1050 1052. 1051 1053. 1052 1054. 1053 1055. 1054 1056. 1055 1057. 1056 1058. 1057 1059. 1058 1060. 1059 1061. 1060 1062. 1061 1063. 1062 1064. 1063 1065. 1064 1066. 1065 1067. 1066 1068. 1067 1069. 1068 1070. 1069 1071. 1070 1072. 1071 1073. 1072 1074. 1073 1075. 1074 1076. 1075 1077. 1076 1078. 1077 1079. 1078 1080. 1079 1081. 1080 1082. 1081 1083. 1082 1084. 1083 1085. 1084 1086. 1085 1087. 1086 1088. 1087 1089. 1088 1090. 1089 1091. 1090 1092. 1091 1093. 1092 1094. 1093 1095. 1094 1096. 1095 1097. 1096 1098. 1097 1099. 1098 1100. 1099 1101. 1100 1102. 1101 1103. 1102 1104. 1103 1105. 1104 1106. 1105 1107. 1106 1108. 1107 1109. 1108 1110. 1109 1111. 1110 1112. 1111 1113. 1112 1114. 1113 1115. 1114 1116. 1115 1117. 1116 1118. 1117 1119. 1118 1120. 1119 1121. 1120 1122. 1121 1123. 1122 1124. 1123 1125. 1124 1126. 1125 1127. 1126 1128. 1127 1129. 1128 1130. 1129 1131. 1130 1132. 1131 1133. 1132 1134. 1133 1135. 1134 1136. 1135 1137. 1136 1138. 1137 1139. 1138 1140. 1139 1141. 1140 1142. 1141 1143. 1142 1144. 1143 1145. 1144 1146. 1145 1147. 1146 1148. 1147 1149. 1148 1150. 1149 1151. 1150 1152. 1151 1153. 1152 1154. 1153 1155. 1154 1156. 1155 1157. 1156 1158. 1157 1159. 1158 1160. 1159 1161. 1160 1162. 1161 1163. 1162 1164. 1163 1165. 1164 1166. 1165 1167. 1166 1168. 1167 1169. 1168 1170. 1169 1171. 1170 1172. 1171 1173. 1172 1174. 1173 1175. 1174 1176. 1175 1177. 1176 1178. 1177 1179. 1178 1180. 1179 1181. 1180 1182. 1181 1183. 1182 1184. 1183 1185. 1184 1186. 1185 1187. 1186 1188. 1187 1189. 1188 1190. 1189 1191. 1190 1192. 1191 1193. 1192 1194. 1193 1195. 1194 1196. 1195 1197. 1196 1198. 1197 1199. 1198 1200. 1199 1201. 1200 1202. 1201 1203. 1202 1204. 1203 1205. 1204 1206. 1205 1207. 1206 1208. 1207 1209. 1208 1210. 1209 1211. 1210 1212. 1211 1213. 1212 1214. 1213 1215. 1214 1216. 1215 1217. 1216 1218. 1217 1219. 1218 1220. 1219 1221. 1220 1222. 1221 1223. 1222 1224. 1223 1225. 1224 1226. 1225 1227. 1226 1228. 1227 1229. 1228 1230. 1229 1231. 1230 1232. 1231 1233. 1232 1234. 1233 1235. 1234 1236. 1235 1237. 1236 1238. 1237 1239. 1238 1240. 1239 1241. 1240 1242. 1241 1243. 1242 1244. 1243 1245. 1244 1246. 1245 1247. 1246 1248. 1247 1249. 1248 1250. 1249 1251. 1250 1252. 1251 1253. 1252 1254. 1253 1255. 1254 1256. 1255 1257. 1256 1258. 1257 1259. 1258 1260. 1259 1261. 1260 1262. 1261 1263. 1262 1264. 1263 1265. 1264 1266. 1265 1267. 1266 1268. 1267 1269. 1268 1270. 1269 1271. 1270 1272. 1271 1273. 1272 1274. 1273 1275. 1274 1276. 1275 1277. 1276 1278. 1277 1279. 1278 1280. 1279 1281. 1280 1282. 1281 1283. 1282 1284. 1283 1285. 1284 1286. 1285 1287. 1286 1288. 1287 1289. 1288 1290. 1289 1291. 1290 12</pre></div></div>

While Loop

A 'while' loop repeatedly executes a block of code as long as a specified condition remains 'True'.

System

1. 1
2. while condition: # Code to repeat

Copy

Example:

1. 1
2. 2
3. count = 0 while count < 5:
4. print(count) count += 1

Copy



© IBM Corporation. All rights reserved.